

Olympiads in Informatics 20

IOI
INTERNATIONAL OLYMPIAD IN INFORMATICS

Olympiads in Informatics

Volume 20, 2026

S. AZIZOVA, S. MURADOV, J. HASANOV	1
Empirical Study of Coding Behavior Under Time Constraints in the International Olympiad in Informatics	
G. FALCÃO, J. CHECCHIA, E. RIBEIRO, M. RAMIRES, M. AMORIM, A. JUNIOR, A. ARAUJO, S. ADI	17
Building Competitive Programmers from Day One: The FACOMpetindo Training Model	
L. MARRONE BERZETTI DI BURONZO	31
Interdisciplinarity as an Entry Point into Computer Science: Women's Pathways, Identity, and Belonging in Hybrid Higher Education	
A. MYKHALKO, O. MITSU, Y. HOROSHKO, H. TSYBKO, V. SHAKOTKO	43
Research on the effectiveness of GPT-4o and GPT-5.4 models in solving olympiad programming tasks on the Eolymp platform	
P. S. PANKOV, S. B. TAGAEVA	61
Tasks in Informatics on Pattern Recognition and Other Senses	
E. SCHRIJVERS	77
Invisible Contributors Task Authors at the IOI and EGOI	
T. D. SERAPIO, ER. J. PANTI, F. E. WU	109
On the Recurrent Themes of Constructives and Interactives in Informatics Olympiads	
Y. TABESH, M. SAGHAFIAN	123
AI in Algorithmic Thinking: Creative Problem Generator	
REPORTS	
M. CROSETTI, A. SANTIAGO GUTIÉRREZ, M. BASSANI, M. MALUF	135
Building a Sustainable Competitive Programming Ecosystem in Argentina: From Olympiad Training to ICPC Institutionalization	
M. DOLINSKY	145
Developing computational thinking for participation in programming olympiads	
T. ITO, S. HIRATA, K. SHIMIZU, H. YONEDA	161
Open Contests in the Japanese Olympiad in Informatics	
M. KAYKOBAD	171
Some Puzzles for Brainstorming	
E. SCHRIJVERS	179
Operational Staffing for an IOI Host: A Practical Role-by-Role Guide	



ISSN 1822-7732

INTERNATIONAL OLYMPIAD IN INFORMATICS

VILNIUS UNIVERSITY

OLYMPIADS IN INFORMATICS

Volume 20 2026

Selected papers of
the International Conference joint with
the XXXVIII International Olympiad in Informatics
Tashkent, Uzbekistan, August 9–16, 2026



Editor-in-Chief

Valentina Dagiene
Vilnius University, Lithuania, valentina.dagiene@mif.vu.lt

Executive Editor

Mile Jovanov
Sts. Cyril and Methodius University, North Macedonia, mile.jovanov@finki.ukim.mk

Technical Editor

Vaidotas Kinčius
Vilnius University, Lithuania, vaidotas.kincius@fsf.vu.lt

International Editorial Board

Benjamin Burton, University of Queensland, Australia, bab@maths.uq.edu.au
Fatih Demirci, Ankara Yildirim Beyazit University, fdemirci@gmail.com
Michal Forišek, Comenius University, Bratislava, Slovakia, misof@ksp.sk
Gerald Futschek, Vienna University of Technology, Austria, futschek@ifs.tuwien.ac.at
Marcin Kubica, Warsaw University, Poland, kubica@mimuw.edu.pl
Luigi Laura, Uninettuno University, Rome, Italy, luigi.laura@uninettunouniversity.net
Ville Leppänen, University of Turku, Finland, villelep@cs.utu.fi
Ágnes Erdősne Németh, Eötvös Loránd University, Hungary, erdosne@inf.elte.hu
Marko Savić, University of Novi Sad, Serbia, marko.savic@dmi.uns.ac.rs
Emil Stankov, Sts. Cyril and Methodius University, North Macedonia,
emil.stankov@finki.ukim.mk
Seiichi Tani, Nihon University, Japan, tani.seiichi@nihon-u.ac.jp

The journal Olympiads in Informatics is an international open access journal devoted to publishing original research of the highest quality in all aspects of learning and teaching informatics through olympiads and other competitions.

<https://ioinformatics.org/page/ioi-journal>

ISSN 1822-7732 (Print)
2335-8955 (Online)

© International Olympiad in Informatics, 2026
Vilnius University, 2026
All rights reserved

Foreword

It is our great pleasure to present the twentieth volume of Olympiads in Informatics. This volume marks a special jubilee for the journal: twenty years of collecting, publishing, and sharing research, reflections, experiences, and practical knowledge from the international informatics olympiad community.

The International Olympiad in Informatics (IOI) is an annual global competition in informatics for individual contestants from more than eighty invited countries. Alongside the contest, the IOI provides a unique meeting place for delegation leaders, organizers, educators, researchers, and guests. The half-day scientific conference held during the IOI offers an excellent opportunity for members of this community to communicate, exchange ideas, report on national initiatives, and discuss emerging challenges in informatics education and competitive programming.

The journal *Olympiads in Informatics* is closely linked to this scientific conference. It focuses on the research and practice of computing professionals who teach informatics to talented secondary and high school students, prepare them for olympiads, design tasks and contests, develop training systems, and study the broader educational and social role of informatics competitions. Over the years, the journal has become a place where both theoretical insights and practical experiences can be shared across national borders.

The twentieth volume continues this tradition. It is organized into two tracks. The first track focuses on research papers addressing questions about contestant behavior, task design, artificial intelligence, interdisciplinarity, inclusion, and the structure of olympiad problem solving. The second track is devoted to reports and national or community experiences, presenting training models, organizational practices, open contests, puzzles, and guidance for future IOI hosts. This volume features contributions from regular authors as well as new contributors, reflecting both continuity and renewal within the IOI community.

The research section opens with a paper by Shahla Azizova, Seyidshah Muradov, and Jamaladdin Hasanov from Azerbaijan, who investigate coding behavior under time constraints in the International Olympiad in Informatics. Their work shows that contest submissions contain rich information beyond final scores. By analyzing structural fingerprints of code and the temporal evolution of submissions, the authors demonstrate that coding trajectories can reveal patterns useful for training, problem design, integrity monitoring, and programming education.

Gabriel Falcão and co-authors from Brazil present the FACOMpetindo training model, an initiative developed at the Faculty of Computing of the Universidade Federal de Mato Grosso do Sul. Their paper describes how competitive programming can be introduced early in university studies, supporting students' development in programming, logical reasoning, and problem solving. The work highlights the importance of systematic training environments and local initiatives in strengthening participation in programming contests.

Laura Marrone Berzetti di Buronzo of Italy examines interdisciplinarity as an entry point to computer science, with particular attention to women's pathways, identity, and belonging in hybrid higher education. Drawing on interviews with female students in hybrid

STEM programs, the paper explores how interdisciplinary pathways can make computer science and competitive environments more accessible. The study contributes to the broader discussion of gender-inclusive learning and the role of institutional recognition in supporting persistence in STEM.

A paper by Andrii Mykhalko, Oleksandr Mitsa, Yurii Horoshko, Hanna Tsybko, and Viktor Shakotko from Ukraine investigates the effectiveness of the GPT-4o and GPT-5.4 models in solving olympiad programming tasks on the Eolymp platform. The authors compare model performance across tasks with Ukrainian and English statements, as well as in C++ and Python. Their findings provide timely evidence on the capabilities and limitations of large language models in olympiad programming, a topic that is increasingly relevant to task setters, teachers, and contest organizers.

Pavel S. Pankov and Sabina B. Tagaeva from Kyrgyzstan propose informatics tasks grounded in pattern recognition and other human “senses,” such as integrity, symmetry, and number sense. Their contribution raises important questions about task design in the age of artificial intelligence. By focusing on problems that rely on human perception and intuition rather than standard algorithmic procedures, the authors invite the community to think creatively about future task design.

Eljakim Schrijvers from the Netherlands contributes a comprehensive catalogue of task authors for the IOI and the European Girls’ Olympiad in Informatics. This paper brings visibility to an essential yet often invisible group: the people who create olympiad tasks. The analysis of task authorship, including the contestant-to-author pipeline and gender representation, offers valuable data for the community and encourages further completion and reflection on the historical record.

Troy Dylan Serapio, EdRonn James Panti, and Farrell Eldrian Wu study recurring themes in constructive and interactive problems across several informatics olympiad archives. Their paper identifies recurring conceptual mechanisms and problem-solving heuristics, showing that such tasks are not merely isolated or ad hoc extensions of standard algorithmic paradigms. This work may help trainers and students better understand the deeper structures underlying constructive and interactive problem solving.

Yahya Tabesh and Morteza Saghafian present a roadmap for an artificial intelligence (AI) system that generates creative multiple-choice problems in algorithmic and computational thinking. Their proposed architecture integrates several AI agents, visual reasoning, a curated knowledge base, and human-in-the-loop validation. The paper addresses a central challenge in informatics education: how to support teachers with scalable tools while preserving originality, educational quality, and meaningful problem-solving value.

The second part of the volume brings together reports and practical experiences from Argentina, Belarus, Japan, Bangladesh, and the Netherlands. These papers focus on building a sustainable competitive programming ecosystem in Argentina; developing computational thinking for participation in programming olympiads in Belarus; organizing open contests in the Japanese Olympiad in Informatics; offering puzzles for brainstorming to develop students’ logic and analytical skills; and providing practical guidance for operational staffing when hosting the IOI.

The volume concludes with a practical guide by Eljakim Schrijvers on operational staffing for an IOI host. Drawing on extensive experience in the IOI community and in organizing major international informatics olympiads, the paper provides a structured, role-by-role overview of the staff and volunteers needed to host the IOI. It outlines responsibilities, timelines, and milestones, offering future hosts a valuable resource for moving from committee formation to successful operational execution.

We extend our deepest gratitude to everyone who contributed to this volume, especially the authors and reviewers. Their dedication and hard work in writing, reviewing, and refining the papers were crucial to creating this collection. We also warmly thank all participants, conference speakers, organizers, and members of the IOI community. We hope that this twentieth volume of *Olympiads in Informatics* will be both a memorable jubilee publication and a useful source of ideas, inspiration, and collaboration for all who care about informatics education and olympiad activity.

Valentina Dagiene

Editor-in-Chief

Empirical Study of Coding Behavior Under Time Constraints in the International Olympiad in Informatics

Shahla AZIZOVA, Seyidshah MURADOV, Jamaladdin HASANOV

School of IT and Engineering, ADA University, Baku, Azerbaijan

e-mail: sazizova11542, smuradov16150, jhasanov@ada.edu.az

Abstract. Competitive programming contests like the International Olympiad in Informatics (IOI) generate rich behavioral data beyond final scores, yet little attention has been paid to systematic analysis of coding styles and code evolution under time constraints. This paper presents an empirical framework for analyzing contestant coding behavior using data from the previous IOI contest. The authors introduce two complementary analytical pipelines: a structural feature extraction system that generates unified structural fingerprints for clustering analysis, and a temporal code evolution tracker that monitors token-level changes and similarity scores across consecutive submissions to reconstruct problem-solving trajectories. The analysis reveals that task structure is the dominant determinant of coding fingerprints, with medal performance associated with a gradient in code richness rather than qualitative stylistic shifts. The authors also present interactive visualization tools enabling coaches and contestants to trace submission histories, distinguish incremental refinements from complete rewrites, and identify behavioral patterns invisible to scoreboard analysis alone. This work contributes actionable insights for contest training, problem design, integrity monitoring, and programming education, demonstrating that systematic examination of how contestants produce solutions reveals meaningful patterns that final scores alone cannot capture.

Keywords: contest, CMS, real-time, coding behavior, stress

1. Introduction

Competitive programming contests provide a unique environment for observing how individuals solve complex algorithmic problems under strict time constraints. Among these contests, the International Olympiad in Informatics (IOI) is widely regarded as one of the most prestigious global competitions for secondary and high school students in computer science. Each year, hundreds of highly talented contestants from around the world compete by solving algorithmic problems that require deep knowledge of data structures,

algorithms, and efficient implementation techniques. While the primary evaluation criterion in such contests is the correctness and efficiency of the submitted solutions, the process through which contestants develop these solutions also contains valuable information about problem-solving strategies and programming practices.

In competitive programming, contestants typically work under strict time limits and high cognitive load. As a result, their coding behavior often reflects pragmatic decisions aimed at maximizing efficiency rather than maintaining conventional software engineering standards. Nevertheless, these coding artifacts provide a rich source of data for understanding how expert-level programmers structure algorithms, organize code, and adapt their implementation styles in a high-pressure environment. Examining these patterns can offer insights not only into competitive programming practices but also into broader aspects of programming education and algorithmic thinking.

Previous studies have explored different aspects of contest data analysis in IOI. In (Alnahhas et al., 2020; Audrito et al., 2025; Kostadinov et al., 2018), authors investigated how contest statistics and operational data can be used to improve contest organization and contestant preparation strategies. Their study demonstrated that analyzing real-time and post-contest data can reveal patterns that help organizers and coaches better understand contest dynamics and participant performance. More recently, (Mammadli et al., 2024; Hasanov et al., 2021) introduced analytical models for evaluating contestant progress and code similarity in real-time coding contests. Their work highlighted the importance of systematic analysis of submissions for monitoring contestant progress and ensuring the integrity of competitive environments. (Lee et al., 2024) analyzes IOI performance data, showing that countries' results correlate with demographic indicators such as population and Human Development Index, but even more strongly with the level of competitive programming interest in the country.

While these studies have demonstrated the value of analyzing contest-related data, relatively little attention has been paid to the systematic evaluation of coding styles and development patterns among contestants. Understanding how participants structure their solutions and evolve their code during the contest may provide valuable insights into algorithmic thinking, efficient programming practices, and competitive programming training methodologies. In particular, examining code organization, use of common templates, implementation patterns, and stylistic characteristics can help identify recurring strategies used by successful contestants.

This study aims to analyze contestant activities and evaluate coding styles in IOI competitions using empirical data derived from contest submissions and development traces. Specifically, the research investigates how contestants structure their programs, what stylistic patterns emerge across different solutions, and how these patterns relate to successful problem solving. The analysis focuses on several aspects, including code organization, naming conventions, modularization practices, and the use of common competitive programming templates. In addition, the temporal sequence of submissions and development iterations is examined to better understand how contestants refine their solutions during the contest.

By systematically analyzing these aspects, the study seeks to contribute to the understanding of programming behavior in high-performance algorithmic environments. The findings may provide insights for educators designing programming curricula, researchers studying algorithmic problem solving, and competitive programming communities interested in improving training methodologies. Ultimately, exploring the coding practices of top-performing contestants can help reveal patterns that characterize efficient algorithmic thinking and rapid software development under time constraints.

2. Problem Statement

The IOI format consists of two competition days (separated with one break day), each containing three algorithmic tasks to be solved within five hours. Contestants submit their solutions to an automated evaluation system, and the final public outcome is typically represented only by the scoreboard, which shows the achieved scores for each participant. While this final result reflects performance, it reveals little about the underlying problem-solving process that leads to the outcome.

In reality, the process of solving algorithmic tasks under strict time constraints involves multiple stages, including planning, coding, testing, debugging, and refining solutions. These stages are reflected in the sequence of submissions and in the evolution of the source code written by contestants. However, this behavioral dimension of competitive programming is rarely visible to observers or researchers because the available public data usually focuses on final scores rather than the development process of solutions.

Previous work has attempted to extract insights from contest activity data. For example, (Hasanov et al., 2021) analyzed submission patterns to understand the strategies of medal-winning contestants based on submission order and frequency. Such approaches demonstrate that submission logs can provide valuable information about the strategies participants employ during contests. Despite these efforts, there is still limited understanding of how contestants differ in their coding behavior under time pressure. In particular, little is known about how participants' coding styles, the evolution of their programs during the contest, and the programming constructs they use may reflect different problem-solving approaches. Understanding these aspects could reveal distinct contestant personas and strategies that are not observable from final scores alone.

Therefore, the problem addressed in this paper is the lack of empirical analysis of coding behavior and code evolution during high-level programming contests. By examining submission histories and source code changes of IOI participants, this study aims to identify patterns in coding style, development dynamics, and programming constructs used under strict time constraints, and to characterize different behavioral profiles of contestants during the competition.

3. Methodology

The study in this paper is performed in two directions: analysis of the coding style of the contestants based on the programming constructs and visualization of the task-based performance to see the dynamics of the code evolution.

The first of the proposed studies has been conceptualized as a multi-stage analytical pipeline that processes the submissions for the programming contests and performs a subsequent analysis and visualization. An overview of the entire workflow is depicted in Figure 1. The colors codes of the first and the second processes are depicted in light brown and green, respectively. As shown in the diagram, the workflow starts with the retrieval of the submissions from the CMS database, which is triggered by the submission control code. Each submission has been considered as an individual unit of analysis. This represents an individual step in the iterative problem-solving behavior of the contestants. After the retrieval process, the source code is subjected to a pre-processing step, in which the code is cleaned and normalized in order to ensure consistency in the input data. As a next step, the pre-processed source code is fed into the syntax analysis component, wherein the code is transformed into an Abstract Syntax Tree (AST). The use of a syntax tree plays a vital role in the methodology, as the structure of the program, as opposed to the surface features of the source code, can be analyzed. Subsequently, the pipeline extracts the features from the AST, yielding a concise set of structural descriptors for the submissions. Finally, the extracted features are subjected to the fingerprint generation module, which transforms the submissions into a unified structural signature.

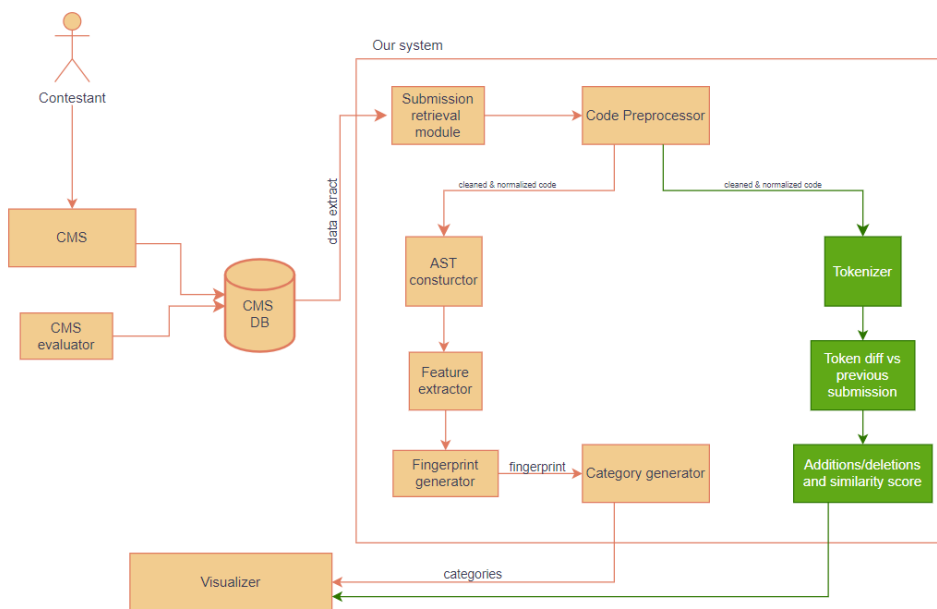


Figure 1. Overview of the Proposed Code Evolution Analysis Pipeline with the structural feature extraction, categorization.

It is noteworthy that the extracted features are transformed into a unified structural signature, allowing for the efficient comparison of the submissions based on the abstraction of the coding style and structure. Following these representations, the category generation module categorizes the contest submissions according to similarities observed in their structural fingerprints, as demonstrated in the flow diagram. These categories represent the implementation patterns and coding behaviors observed among the contestants. Lastly, the results are displayed through a visualization module, which allows for the interpretation of the observed structural patterns at different levels of aggregation, from individual contest submissions, tasks, and finally, contestants. This is in line with the objective of identifying the differences in the approaches adopted by the contestants in solving the problems and their coding styles. Overall, as demonstrated in Figure 1, the pipeline represents a systematic transformation from the raw contest submissions to the results, with a clear emphasis on modularity, reproducibility, and the structure of the programs as opposed to their programming characteristics.

3.1 Structural Analysis Pipeline Implementation

As mentioned earlier, the structural analysis pipeline is implemented through automated scripts that process raw source code submissions to extract structural representations from them. Each script within the process chain completes particular computations and outputs actual results, ensuring that the whole procedure can be replicated. The process starts with processing each source code file received in the course of the coding competition, where every file corresponds to one submission. Within the framework of the chosen approach, each submission will be analyzed individually, providing opportunities to examine the task and author-related features. In the initial step, the input code is subject to normalization. During normalization, non-structural elements such as comments, redundant whitespace, and formatting variations are removed, and this ensures consistency across inputs and is stored in the *normalized_code* file, associated with the unique submission. A copy of the code in its original form is saved under the file name *raw_submitted_code* and can serve for further referencing if needed. Following normalization, an Abstract Syntax Tree (AST) is generated using the Tree-sitter framework (Sklower, M. 2018). The AST represents the hierarchical syntactic structure of the source code, enabling analysis at the structural level rather than relying on raw text. Finally, the created abstract syntax tree is translated into the JSON format and saved in the *ast.json* file associated with the unique submission. Each stage of the pipeline yields a structure that is stored in a separate file with the reference to the unique submission. This approach provides end-to-end traceability from the original inputs to their final output forms enabling individual examination and validation of each step separately.

Based on the AST, several structural attributes are calculated to provide quantitative analysis of the code. These attributes cover various aspects of software development, such as structural complexity and flow control structures. The following list shows the calculated attributes:

- Maximum depth of the AST (proxy for structural complexity)
- Number of functions
- Number of loops and conditional statements
- Number of return statements
- Counts of control flow modifiers such as break and continue
- Number of included libraries and macro definitions

The calculated attributes are saved in two formats – as a CSV file `features.csv` and JSON file `features.json`.

3.2 *Feature Vector Construction and Fingerprinting*

For the purpose of comparing all submissions efficiently, the extracted features are converted into a feature vector with a fixed order. The order of the features is specified explicitly to maintain uniformity while comparing features. The feature vector is serialized into a concise format (for example, “11|2|0|4|6|0|0|2|0”) and acts as a signature of the structural details of the program. A SHA-256 hash function is applied to this string to create a fingerprint of the structure of each submission. The fingerprint is saved in `fingerprint.json` along with the original feature vector and feature order. The above operation facilitates comparison between submissions, enables the identification of structurally similar programs, and supports grouping of coding styles.

3.3 *Feature Selection and Representation*

In terms of choosing the structural features, the objective is to represent each program in such a way that sensitivity to surface variation is minimized while preserving key structural characteristics of the program. In this study, the analysis does not rely directly on the raw source code, but rather on features derived from its AST. The reason behind this decision is to ensure that the representation is independent of surface variations such as formatting or individual programming style. The features consist of counts of fundamental program structures, such as functions, loop constructs, and conditional statements. The attribute set deliberately does not contain any elements that can only be obtained based on specific naming schemes, indentation styles, etc. This ensures that structural information about the programs written by various contestants can be represented in a uniform manner. It should not be assumed that these attributes describe the full range of coding behavior exhibited by the participants. Instead, they represent a simplified model of structural information about their work. For purposes of comparison, all the extracted features are put into an ordered vector. This way, every submission will have its features in a uniform form so that comparisons can be made among various programs. This vector is then converted to string form and hashed using a normal cryptographic algorithm. The hashing process serves merely the purpose of creating unique identifiers for each structural feature set and does not make any assumptions beyond this.

3.4 Code and performance evolution measurement

The second study pursues a different correspondence pipeline, which involves monitoring the dynamic progress of individual contestants regarding the dynamic development of their code in successive submissions of a particular task. As demonstrated in (Kostadinov et al., 2018), analyzing the sequence of submissions rather than only the final score reveals behavioral patterns that are not observable from the scoreboard alone. An overview of the proposed workflow is depicted in Figure 1.

Similar to the first study, the pipeline is driven by submissions retrieved from the CMS database. Each submission is uniquely identified by its submission ID, contestant identifier, task name, and the score assigned by the automated evaluation system. Submissions are grouped by contestant and task, then ordered chronologically by submission ID to reconstruct the development timeline of each solution.

After the submissions are ordered, every source file continues to a pre-processing phase. Single line comments, multi-line comments and preprocessor directives are eliminated. Tab characters are converted to single spaces, successive spaces are merged, and all empty lines are removed. This step is used in order to make sure that the later structural analysis is not influenced by superficial variations in the formatting or commenting style which range widely across contestants, but which do not carry algorithmic significance. Table 1 illustrates the transformation applied to a representative code fragment across the three stages: original source, cleaned source, and normalized token sequence.

The processed source code is then subjected to the tokenization module which converts the processed source code into a normalized sequence of abstract tokens. Inspired by established token-based structural analysis approaches (Prechelt et al., 2002), all identifiers and keywords are replaced with the generic token *ID*, all numeric literals are replaced with *NUM*, and operators are retained in their original form. This normalization ensures that two structurally identical solutions written by different contestants but using different variable names produce identical token sequences, enabling a fair structural comparison. After tokenization, pairs of subsequent submissions of the same contestant and task are compared by using a diff function directly on the token sequences. The *diff* calculates the number of tokens that were added and removed between the current and the previous submissions, which gives a quantitative measure of the extent to which the code structure was modified at each step. As a by-product of the same comparison, a similarity score is calculated with the Ratcliff-Obershelp algorithm (Ratcliff et al., 1988) and results in a ratio of 0 to 100 percentage, with 100 percentage being a structurally identical submission and lower values reflect a more substantial reorganization of the solution.

Table 1. Example of the three-stage pre-processing transformation applied to a contestant’s source code.

Original code	Cleaned code	Token sequence
<pre>#include <bits/stdc++.h> #define MAX 100 /* initialize sum */ int n, sum = 0; // loop over input for(int i=0;i<n;i++){ sum=sum+a[i]; // add }</pre>	<pre>(removed) (removed) (removed) int n, sum = 0; (removed) for(int i=0;i<n;i++){ sum=sum+a[i]; }</pre>	<pre>ID ID , ID = NUM ; ID (ID ID = NUM ; ID < ID ; ID ++) { ID = ID + ID [ID] ; }</pre>
8 lines · 4 comments/directives	4 lines · comments removed	19 tokens · 3 types: ID, NUM, operators

Finally, the results are passed to the visualization module which shows the progress of the selected contestant on the given task over the time. The progress contains the additions/removals to the code at each submission, with the received score.

4. Experiments

4.1 Dataset and Experimental Setup

The proposed models have been run on 8113 submissions from International Olympiad in Informatics 2020 hosted in Singapore, held online from 13–19 September 2020, which gathered 343 contestants from 87 countries and awarded 29 gold, 57 silver, and 85 bronze medals. Each record in the fingerprint table corresponds to a single (contestant, task) pair and is identified by a composite key of the form *contestant_taskname*. From each submission, nine structural features were extracted via abstract syntax tree (AST) analysis: (1) *ast_max_depth*, the maximum nesting depth of the parse tree; (2) *num_functions*, the number of distinct function definitions; (3) *num_loops*, the total count of loop constructs; (4) *num_conditionals*, the number of conditional branching statements; (5) *num_returns*, the count of return statements; (6) *num_breaks*, the count of break statements; (7) *num_continues*, the count of continue statements; (8) *include_count*, the number of preprocessor include directives; and (9) *macro_count*, the number of macro definitions. Together, these nine dimensions constitute what we term the *coding_fingerprint* of a submission, capturing structural and stylistic properties of a contestant’s solution independent of correctness.

4.2 Data Processing and Cluster Analysis

Prior to analysis, the fingerprint table was joined with the medal registry on the contestant identifier extracted from the composite key. All nine features were normalized by their respective observed maxima across the entire dataset to produce values in the range [0, 1], enabling cross-feature comparison on a common scale. To identify latent groups of coding behavior, we applied *k*-means clustering (Lloyd’s algorithm, *k* = 6, 20 random initializa-

tions, random seed 42) to the standardized (zero-mean, unit-variance) feature vectors. The number of clusters was selected to balance interpretability with granularity; preliminary runs with $k \in \{4, 5, 6, 7\}$ confirmed that six clusters yielded the most semantically coherent groupings without over-segmenting the data. Cluster centroids were back-transformed to the normalized scale for visualization. To characterize coding styles across outcome groups and tasks, we additionally computed per-group mean normalized profiles for the four medal tiers and the three tasks separately. All profiles were visualized as nine-axis radar charts, with each axis corresponding to one of the nine extracted features.

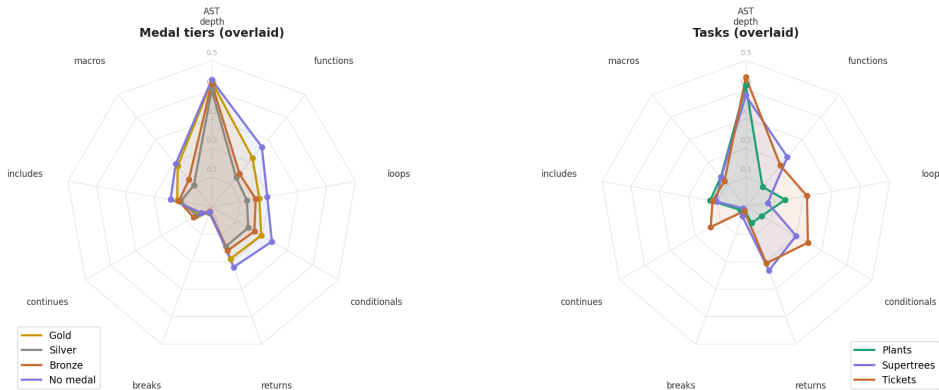


Figure 2. Generalized coding fingerprint profiles per medal tiers and tasks.

The radar profiles for the four medal tiers (Figure 2a) reveal a consistent but moderate gradient across all nine features: Gold medalists exhibit higher mean values than Silver, Bronze, and unmedalled contestants on *ast_max_depth*, *num_functions*, *num_conditionals*, and *num_returns*, while the pattern on loop-related and flow-control features is less discriminative. Crucially, the shape of the radar polygons is broadly concentric - the qualitative structure of a submission’s fingerprint does not change categorically between medal tiers; rather, the magnitudes scale monotonically with performance. This suggests that higher-performing contestants write structurally richer code, particularly with respect to functional decomposition and branching complexity, rather than adopting a qualitatively different programming style.

The task-level profiles (Figure 2b) present a strikingly different picture. The three tasks impose markedly distinct structural demands: *supertrees* elicits deep AST structures and high loop and continue counts, reflecting the iterative graph-theoretic nature of the problem; *plants* rewards functional decomposition, with the highest mean *num_functions* and *num_returns* across all three; and *tickets* yields the sparsest profiles overall, with notably low conditional and return counts, consistent with its more algebraic character. These task-level differences in fingerprint shape are substantially larger than the inter-tier differences within any single task.

The same clustering framework can be naturally extended to the individual level by mapping each contestant’s submissions onto the derived clusters over time. This enables track-

ing how a specific participant’s coding behavior evolves, allowing a coach to compare their structural patterns, consistency, and progression against those of other team members or against aggregate profiles. Such an analysis could provide actionable insights for targeted training and performance improvement. However, due to the increased complexity of visualizing individual trajectories alongside cluster-level summaries, as well as considerations related to participant privacy, we deliberately chose not to include identifiable individual-level results in this study.

Table 2. Summary of the six identified coding style clusters. n = number of (contestant, task) pairs assigned to each cluster.

Cluster	Label	n	Dominant task	Top medal tier
1	Loop-heavy	246	Supertrees (79%)	Bronze / None
2	Break-intensive	50	Mixed	None (68%)
3	Macro-rich	145	Plants + Tickets	Balanced
4	Function-first	84	Plants (71%)	Balanced (Gold 25%)
5	Minimal / sparse	430	Plants + Tickets	None (59%)
6	Include-heavy elite	8	Mixed	Gold + Silver (75%)

The six clusters (Table 2) reinforce the primacy of task structure. Cluster 1 (Loop-heavy, $n = 246$) is composed almost entirely of supertrees submissions (79%) and is characterised by elevated loop and continue counts. Cluster 5 (Minimal/sparse, $n = 430$, the largest cluster) captures the majority of plants and tickets submissions written with flat, structurally simple code; its medal distribution skews heavily toward unmedalled contestants (59%). Cluster 4 (Function-first, $n = 84$) is the most medal-predictive group: with 25% Gold, 29% Silver, and 24% Bronze assignments, it is the only cluster with a near-uniform medal distribution, suggesting that a habit of functional decomposition — as applied predominantly to the *plants* task — correlates with competitive success independently of task assignment. Cluster 6 (Include-heavy elite, $n = 8$) represents a small population of experienced contestants who employ extensive preprocessor machinery (mean normalised *include_count* ≈ 0.87 , *macro_count* ≈ 0.41); 75% of its members are Gold or Silver medallists, consistent with the use of competitive-programming template headers as a marker of prior experience.

Cluster 2 (Break-intensive, $n = 50$) is notable for its disproportionately high *num_breaks* score (mean ≈ 0.22 , more than ten times the dataset mean), despite otherwise unremarkable profiles; 68% of members are unmedalled. While the causal direction is unclear, the elevated break usage may reflect iterative or brute-force solution strategies that do not scale to full marks under IOI constraints.

A web-based application has been developed to trace the performance of the individual contestant. The dashboard of the application enables the selection of the contestant and the task from the list (Figure 3) and see the performance of the user over the time.

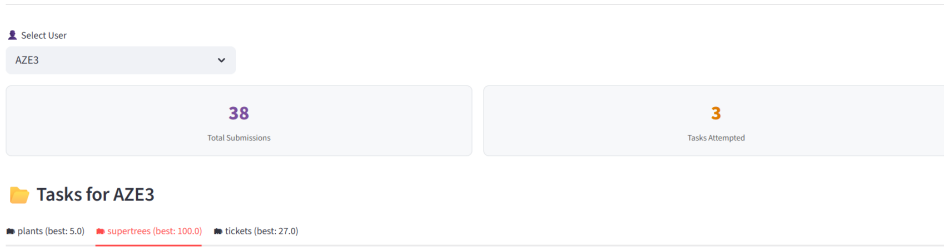


Figure 3. A visualization tool for tracing the contestant performance

Figure 4 shows the token addition/deletion graph of the same contestant in a series of submissions on a given task. The green bars indicate tokens added and the red bars indicate tokens removed on each submission step and the dotted orange line indicates the corresponding score trajectory on the right axis. Figure 5 shows a token-based similarity chart of that same contestant, which reveals how structurally similar each submission was to the previous submission. Combining these two visualizations enables the viewer to trace the structural code changes with performance improvements or decreases, and to draw a line between individual refinements and total rewrites of the solution. Overall, the second pipeline transforms raw submission histories into an interpretable representation of the problem-solving process, enabling the analysis of development patterns, revision strategies, and score progression at both the individual contestant and task levels.

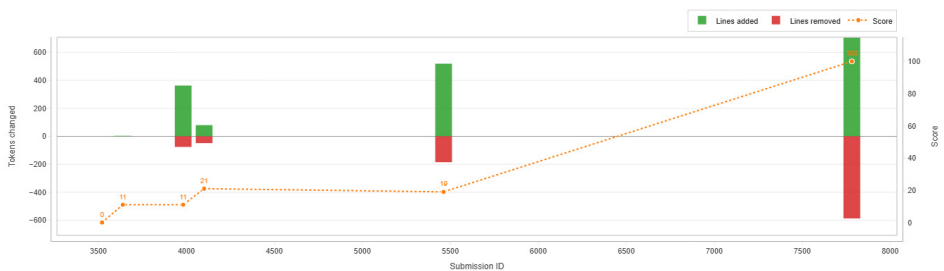


Figure 4. Token Additions and Deletions Across Consecutive Submissions with Score Trajectory for a Representative Contestant.

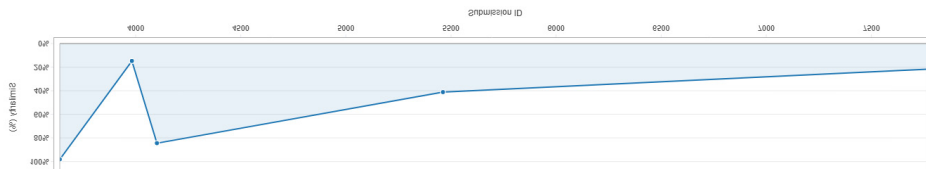


Figure 5. Token-Based Structural Similarity Between Consecutive Submissions for a Representative Contestant.

5. Conclusion

In this study, we have presented an empirical framework for analyzing coding behavior under time constraints in the International Olympiad in Informatics, introducing two complementary analytical pipelines: structural feature extraction for code fingerprinting and temporal code evolution tracking across submission sequences. Our methodology demonstrates that systematic examination of contestant submissions reveals meaningful patterns beyond final scores, offering valuable insights into problem-solving dynamics in high-pressure algorithmic environments. By shifting the focus from final scores to the underlying development process, the proposed multi-stage pipeline provides a granular view of how contestants navigate complex algorithmic challenges under strict time constraints and high cognitive load. The experimental application to 8,113 submissions from IOI 2020 demonstrates that our approach successfully identifies structural patterns and enables the objective grouping of coding styles across contestants from different countries and backgrounds. The provided visualization of changes alongside score trajectories allow observers to distinguish between incremental refinements and complete solution rewrites, thereby illuminating the strategic decisions contestants make when confronted with failing submissions or time pressure.

Our findings carry substantial practical implications for multiple stakeholders within the competitive programming community. For contestants, our analytical framework offers opportunities for systematic self-reflection and skill development. By examining their own submission histories, contestants can identify recurring patterns in their coding behavior, such as whether they tend to iterate gradually through small modifications or restructure solutions entirely after encountering errors. Understanding these personal tendencies can help contestants develop more effective revision strategies, improve time management during competitions, and ultimately enhance their performance. Moreover, comparing one's structural fingerprints with those of higher-performing peers may reveal algorithmic patterns or coding practices that warrant adoption. For coaches, our system provides a diagnostic tool that extends far beyond traditional scoreboard analysis. Rather than relying solely on final outcomes, coaches can examine how their students' code evolves over the course of a contest, identify recurring inefficiencies such as excessive restructuring or frequent minor fixes that indicate inadequate initial planning, and tailor training interventions accordingly. The ability to visualize token addition and deletion patterns alongside similarity scores enables coaches to deliver evidence-based feedback on problem-solving approaches, helping students internalize more efficient coding practices under competition conditions. For the IOI technical and scientific committees, our work contributes several actionable outcomes of both operational and strategic significance. The fingerprinting mechanism can enhance contest integrity by enabling the automated detection of unusually similar solution structures across contestants, potentially identifying unauthorized collaboration or code sharing. The evolution analysis can inform problem design and contest formatting: tasks that consistently produce erratic submission patterns, low structural stability, or high rates of complete rewrites may indicate poor problem specification, ambiguous statement wording, or unexpected difficulty spikes. Furthermore, our pipeline provides

a replicable framework for post-contest analysis, allowing committees to generate detailed behavioral reports that go beyond traditional scoreboards and medal distributions, thereby informing evidence-based decisions about contest duration, time allocation across tasks, and evaluation criteria.

The results of the cluster analysis lead to three principal conclusions. First, task structure is the dominant determinant of a contestant's coding fingerprint. The algorithmic demands of each problem, specifically graph traversal, recursive decomposition, or arithmetic reduction impose structural constraints on solutions that overshadow the effects of individual skill level. This has methodological implications for studies that seek to use static code features as proxies for programmer ability: any such analysis must control for problem identity to avoid confounding. Second, medal performance is associated with a gradient in code richness, not a qualitative shift in style. Gold-tier contestants write structurally deeper and more decomposed code, but the overall shape of their fingerprints is not categorically different from lower-tier contestants solving the same task. This gradient is most pronounced in function count and return statements - features that proxy for modular design and least pronounced in loop and flow-control features, which are more strongly determined by task requirements. Third, a function-first coding style is the strongest per-cluster predictor of medal success. Cluster 4, characterized by high function decomposition and applied predominantly to the plants task, is the only cluster in which Gold, Silver, and Bronze members are represented at approximately equal rates. This finding is consistent with the pedagogical literature linking structured, modular programming to problem-solving efficacy in competitive contexts. Taken together, these findings suggest that static structural fingerprints capture a mixture of task-specific and contestant-specific signal. Disentangling these contributions, for instance, through within-task normalization or multi-task fingerprint aggregation remains an important direction for future work.

We also see several promising avenues for future work emerging from our study. Expanding the feature set to include more sophisticated metrics, such as cyclomatic complexity, information-theoretic measures of code entropy, or dependency graph analysis, could provide deeper insights into the algorithmic design choices contestants make under time constraints. Applying our pipeline to multiple IOI editions would enable longitudinal studies of how coding styles and problem-solving strategies evolve across years and in response to changes in contest format or available programming languages. Integrating machine learning techniques could enable predictive models that identify contestants at risk of poor performance based on early submission patterns, potentially facilitating real-time interventions in training environments. Cross-contest comparisons between IOI and national contests or online judges could reveal how contest format and time pressure influence coding behavior across different contexts. Finally, developing an interactive web-based platform that makes our analytical tools accessible to the broader competitive programming community would maximize the practical impact of this work, allowing coaches, contestants, and researchers to explore submission data interactively.

In summary, our study demonstrates that systematic analysis of coding behavior in competitive programming contests reveals meaningful, actionable patterns that are entirely

invisible when examining final scores alone. The frameworks we have introduced for structural fingerprinting and evolution tracking provide valuable tools for understanding algorithmic problem-solving under time constraints, with direct applications for training methodologies, contest design, performance evaluation, and integrity assurance. As competitive programming continues to grow in global prominence, we believe that the ability to analyze not merely what contestants produce but how they produce it will become increasingly important for educators, coaches, organizers, and researchers alike.

References

- Alnahhas, A., Mourtada, N. (2020). Predicting the performance of contestants in competitive programming using machine learning techniques. *Olympiads in Informatics*, 14, 3–20. DOI: 10.15388/ioi.2020.01
- Audrito, G., Laura, L., Orlandi, A., Ostuni, D., Rizzi, R., Versari, L. (2025). Interactive problem solving in the classroom: Experiences with Turing Arena Light in competitive programming education. *Olympiads in Informatics*, 19, 1–25. DOI: 10.15388/ioi.2025.01
- Hasanov, J., Gadirli, H., Baghiyev, A. (2021). On using real-time and post-contest data to improve the contest organization, technical/scientific procedures and build an efficient contestant preparation strategy. *Olympiads in Informatics*, 15, 23–36. DOI: 10.15388/ioi.2021.03
- Kostadinov, B., Jovanov, M., Stankov, E. (2018). Platform for analysing and encouraging student activity on contest and e-learning systems. *Olympiads in Informatics*, 12, 85–98. DOI: 10.15388/ioi.2018.07
- Lee, E., Reizin, T., Wu, F.E., Wu, F.E. (2024). Trends on returning contestants and geography at the International Olympiad in Informatics. *Olympiads in Informatics*, 18, 33–50. DOI: 10.15388/ioi.2024.0333
- Mammadli, M., Mammadli, N., Hasanov, J. (2024). Analysis and evaluation of the contestant’s progress in real-time coding contests. *Olympiads in Informatics*, 18, 51–62. DOI: 10.15388/ioi.2024.0451
- Prechelt, L., Malpohl, G., Philippsen, M. (2002). Finding plagiarisms among a set of programs with JPlag. *Journal of Universal Computer Science*, 8(11), 1016–1031. DOI: 10.3217/jucs-008-11-1016
- Ratcliff, J.W., Metzener, D. (1988). Pattern matching: The Gestalt approach. *Dr. Dobb’s Journal*, 46.
- Sklower, M. (2018). Tree-sitter: An incremental parsing system for programming tools. Tree-sitter. Available at: <https://github.com/tree-sitter/tree-sitter>



S. Azizova is a Computer Science graduate from ADA University. Her interests include data analysis, machine learning, and computer vision applications. She is currently working on different data analytics tasks that analyzes the computer code and biological data sequences.



S. Muradov is a last year Information Technology student at ADA University. His interests include data analysis and machine learning. As a part of his Senior Design Project, Seyidshah is using CMS and the contest data to design a system for the code progress review and analysis.



J. Hasanov is an Associate Professor of Computer and Information Sciences at ADA University. Dr. Hasanov is mainly focused on image processing and machine learning problems covering text and digital object recognition domains. Additional to the research field, Dr. Hasanov teaches the management aspects of the IT in production and operation environments. Dr. Hasanov has been an ITC member for the period of 2017-2020 and chaired HTC during IOI 2019.

Building Competitive Programmers from Day One: The FACOMpetindo Training Model

Gabriel FALCÃO¹, Jenniffer CHECCHIA¹, Erick RIBEIRO¹,
Mariana RAMIRES¹, Matheus AMORIM¹, Amaury JUNIOR¹,
Anderson ARAUJO¹, Said ADI¹

¹Faculdade de Computação - Universidade Federal do Mato Grosso do Sul Campo Grande, MS - Brazil

e-mail: {gabriel_falcao, jennifer_checchia, erick_rodrigues, mariana.ramires, matheus.amorim, amaury.junior, anderson.vicoso, said.sadique}@ufms.br

Abstract. Programming, problem-solving, and logical reasoning skills are fundamental to student’s academic and professional development. In Brazil, competitions such as the *Olimpíada Brasileira de Informática* (OBI) and the *Maratona Feminina de Programação* (MFP) foster these competencies within the field of competitive programming.

In this context, this work presents FACOMpetindo, an initiative at the Faculty of Computing of the *Universidade Federal de Mato Grosso do Sul* (UFMS), in the state of *Mato Grosso do Sul*, Brazil, aimed at disseminating competitive programming, training students and strengthening participation in a number of contests.

Keywords: competitive programming, freshmen training, programming contests, educational methodology, Brazil.

1. Introduction

Competitive programming is classified as a Mind Sport [Nunes 2022], which means an intellectual competition focused on improving cognitive abilities. The programming tests aim to incentive problems’ resolution in a certain period of time, using algorithm’ knowledge and programming techniques. Between brazilian competitions, stand out the *Maratona de Programação da Sociedade Brasileira de Computação* (SBC Marathon, that can be translated as “Computing Brazilian Society’ Programming Marathon”), the *Olimpíada Brasileira de Informática* (OBI, or “Brazilian Informatics Olympiad”) and the *Maratona Feminina de Programação* (MFP, or “Feminine Programming Marathon”).

SBC Marathon, annually organized by the SBC, has been ongoing since 1996 and is intended to undergraduation and graduation students of computer-related areas. The competition evolve regional stages, where the best teams are qualified to the following stages, reuniting students of all around the world into the global stage from International Collegiate Pro-

gramming Contest (ICPC). SBC Marathon aims to promote participants' teamwork, logic thinking and creativity through problem resolution on different difficulty levels. The teams are composed by a coach (can be a professor) and three competitors (must be students), plus one, as a substitute; they shall solve the biggest amount of problems at the shortest time and with less penalties as possible¹.

In 2025, the Phase One (Regional Phase) of the Marathon counted on 1025 participating teams, presenting 239 institutions, realized simultaneously with other countries of Latin America, such as Bolivia, Mexico and Central America countries. In the Brazilian regions, the *Centro-Oeste* had the 2nd smaller participation, surpassing only the *Norte*. As a consequence, the participant's proportion on the Final Phase followed a similar pattern, with the *Sudeste* as a leader (49,2%), followed by *Nordeste* (18,5%), *Sul* (13,8%), *Centro-Oeste* (12,3%) and *Norte* (6,2%) as presented on Figure 1.

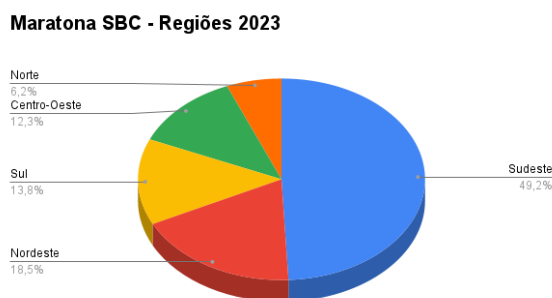


Figure 1. Regional participation graphic on 1st phase of Maratona SBC 2023

The OBI is focused on Middle and High School students (but also accepts from technical education and entrants in undergraduation), while the Marathon has 2 subscription categories and just 1 one them if for actual competitors (whereas the other is for practicing only), the OBI is divided in 2 different competition modalities²:

1. The introductory category consists of a test with logic and computer problems (programming logic, but without the use of a computer). This level is intended for elementary school students and is subdivided into: Junior Level, for fourth or fifth grade students; Level 1, for sixth or seventh grade students; and Level 2, for eighth and ninth grade elementary school students.
2. The programming category consists of a test with programming tasks that must be solved using a computer, much like the IOI. This modality is intended for elementary and high school students who have prior knowledge of some programming language. This modality is subdivided into Levels according to schooling: Junior Level, for students of any year of elementary school; Level 1, for students up to the first year of high school; Level 2, for students up to the third year of high school; and Senior Level, for

¹<https://maratona.sbc.org.br/>

²<https://olimpiada.ic.unicamp.br/>

students who are in their fourth year of technical school or who are, for the first time, in their first year of an undergraduate course.

Both Marathon and OBI foster girl participation by offering exclusive competitions. That's the case of the mentioned "MFP" and the feminine category of OBI. These have the objective of increasing female and non-binary people participation at competitive programming, which is usually low compared to male participation [Nunes et al. 2024, Silva et al. 2023]. The most impactful one being the MFP, incentivizing and covering up students from all country in a safe and welcoming event, composed of two stages, regional and final in "UNICAMP"³.

The participation in competitive programming events, as the mentioned ones, contribute to increase skills such as: problem solving; teamwork; innovation; creativity and computational thinking, which are fundamental abilities for students and professionals in the 21st Century [R. Raman and Achuthan 2018, Moura A. F. et al. 2019]. Therefore, the people who dedicate part of their time to competitive programming are the most probable to take up great positions in big companies such as "Google", "Amazon" and "Microsoft" [Nair 2020, WAGEED et al. 2024].

A well-documented challenge in STEM education and careers is the lack of diversity [Verdugo-Castro et al. 2022], reaching proportions such as 16% to 20% of women in the computational graduation courses. In 2023, the SBC Programming Marathon started directing about 10% of the slots for the Final Phase to teams with feminine competitors, improving the participation in 200% on this phase, and about 15% on 2024. This action also increased the first phase participation in 30% [Ramos et al. 2025], which demonstrates the necessity of female inclusion.

Facing the different approaches used for boosting competitive programming, we can observe that Quintela, Araújo and Pereira utilize a learning method applied to basic and high school classes that evolves weekly encounters, study trails on the course "*Programação Básica (Cod Cad) - De Zero a Herói*" (Basic Programming Cod Cad From Zero to Hero), focusing on C++. As a result, the methodology is structured focusing on competitive programming since high school. [Quintela and Ribas 2016]

On a related work, [Vianna et al. 2020] presents a mini course focused in construction and development of strategies on math and logical problem solving on different difficulty levels. The mini course relies on encounters and discussions about solutions based on previously elaborated strategies, integrating math and computing into competitive programming.

Finally, [Brito et al. 2019] applied, coupled with students of Computer Science at *Universidade Federal de Ouro Preto*, a methodology that combines gamification and online exercise platforms to incentive participation.

On this context, this work stands out by the production of online didactic material, weekly training, event organization and incentive *FACOM/UFMS's* student participation on olym-

³<https://www.instagram.com/mfp.sbc/>

piads and marathons. Those actions are conducted by a support network composed by volunteer professors and students. Thus, this work presents an experience report of the project *FACOMpetindo* since its creation on 2024, being focused on freshman from Computer related areas (Computer Science and Engineering, Software Engineering, Information Systems and Artificial Intelligence) on competitive programming.

The results include the developed materials and graphics that demonstrate the increase on student engagement on those competition. In the future, a deeper study will validate the project's impact on their course's motivation, as well as success rate and evasion.

2. Methodology

The main challenge in developing a training methodology, is the relative short span of a academic semester, where the students, specially the freshmen, also have to study for the other standard college subjects. Also, the new students at FACOM learn exclusively Python in their first semester. In this context, the methodology was divided in 3 parts, described below:

Trainings: The trainings are taught in every academic semester, occurring weekly in two distinct schedules, one of them in the morning and the other in the afternoon, despite this division, the same content is presented. The reasoning behind this separation is to achieve the maximum amount of interested students as possible, lessening schedule conflicts with the college subjects.

The chosen language for the trainings was C++ for being the most used language in the world for competitive programming. In light of the fact that the majority of the target audience are new students and that they learn exclusively Python in the beginning, it is necessary that the lessons start at the introduction of the language, explain the syntax and programming fundamentals like loops, if-else, variables and arrays. The remaining algorithms, techniques, and data structures taught are based on the OBI syllabus⁴ and other works [Manev 2024]. In the table 1 is presented the final syllabus planned and applied in the trainings of 2025/1.

Table 1. FACOMpetindo's Training Timeline 2025.1

Week	Content
01	Introduction to C++, Algorithm complexity
02	Sublime configuration, Loops, Conditionals
03	Vectors, Matrix, Strings
04	Prime Numbers, Euclides, Binary Search
05	Bits, Functions, Recursion
06	Ad-hoc, Greedy Algorithm
07	Test
08	STL e Union-Find

⁴https://olimpiada.ic.unicamp.br/prepare/ementas/ementa_prog/

09	Graphs, BFS, DFS
10	Dynamic Programming
11	Ending

As a supplementary training tool, a Training Spreadsheet was developed⁵, as shown in Figure 2. In this spreadsheet, a new tab was created each week with suggested exercises from the Codeforces and Beecrowd platforms to complement and practice the content covered. Furthermore, a kind of gamification was applied, where students enter their name and which exercises they managed to solve (AC), had an incorrect solution (WA), or had an inefficient solution (TLE) for the other students to see. A hint was also included for each problem, so that students could read it if they couldn't solve a problem initially.

The suggested exercises are analyzed and solved by volunteer students from the project. For each topic covered, some exercises are chosen that can be solved using the acquired knowledge, and many others that do not require the week's content to be solved, so that the progression between topics is smooth. All solutions are available online at <https://github.com/FACOMpetindo/beecrowd>.

The simulation was conducted on the ThemeCP platform, a website created to strengthen the practice of exercises in Codeforces, creating personal and customized contests. The closing ceremony includes gifts and some prizes for the students, such as the Fibonacci Award (most exercises solved) and the While (true) Award (highest attendance), to encourage and congratulate the students.

	A	B	C	D	E	F	G	H	I
1	SLIDES			Fácil			Médio		Difícil
2	Names:	Texto Bagunçado	Score	sensor diferente	Cogumelos	Jogo de Basquetebol	Apaixone	Verify password	O mais rápido
3	DICAS ->	DICA	DICA	DICA	DICA	DICA	DICA	DICA	DICA
4	Miguel Bernal	AC	AC			AC	TLE		
5	Mateus Sposo	AC	AC	AC	AC		TLE		
6	Cássio Q Minozzo	AC	AC	AC	AC				
7	João Pedro Pinheiro	AC	AC	AC					
8	Ricardo Consolário			AC	AC	AC			
9	Rodrigo Kenji								
10	João Marcelo	AC	AC	AC					
11	Kaui Alves	AC	AC						
12	Larissa de Lima Santos	AC							
13	Yan Leandro R de Oliveira	AC							
14	João Pedro Huppel	AC	AC	AC					
15	Ana Clara M. Silva	AC							
16	Mateus Zeni	AC							
17	Daniely Maximo Marconi		AC						
18	Matheus Rocha	AC	AC	AC	AC				
19	Otávio Gabriel de Oliveira								
20									
21									
22									

Figure 2. Training Spreadsheet

Online Study Material: An online study material was developed⁶ as well, using the *Gitbook*⁷ platform, to facilitate asynchronous learning, targeting students who cannot attend the in-person trainings. The content presented aims to replicate what is covered in class, with code in both Python and C++. At the end of each section, the reader is provided with some exercises from the Beecrowd platform related to the topic discussed.

⁵https://docs.google.com/spreadsheets/d/1kXiKp2OSLasikJap7iRK_

⁶<https://facompetindo.gitbook.io/programacao-competitiva>

⁷<https://www.gitbook.com/>

Event organization: Another key point of the FACOMpetindo methodology is organizing events whenever possible, whether these are the local headquarters for official competitions or their own events. An example of the latter is the *Esquentando FACOMpetindo* (FACOMpetindo Warm-up), an event held for the first time in 2025 in an online format. The aim of this event was to create a real but accessible competition, with exercises selected by project members and targeting undergraduate students in the state.

The first edition of the FACOMpetindo Warm-up had 18 competitors⁸, 17 from UFMS and 1 from UCDB, another college in the same city. In 2026, the second edition of the FACOMpetindo Warm-up is planned, but this time in a face-to-face format.

3. Participation in Competitions

By applying the methodologies described above, an increase in curiosity and interest in participating in events related to competitive programming was observed. The following describes the frequency and results achieved by FACOM students in the biggest competitions in Brazil.

Since the project was structured in 2024, the participation of FACOM students in the OBI has been increasing significantly. In 2025, phase one of the OBI had more than 40 freshmen representing the faculty, out of a total of 56 registered, which represents a new record in both participation and registrations; of these, 10 students achieved perfect scores on the exam.

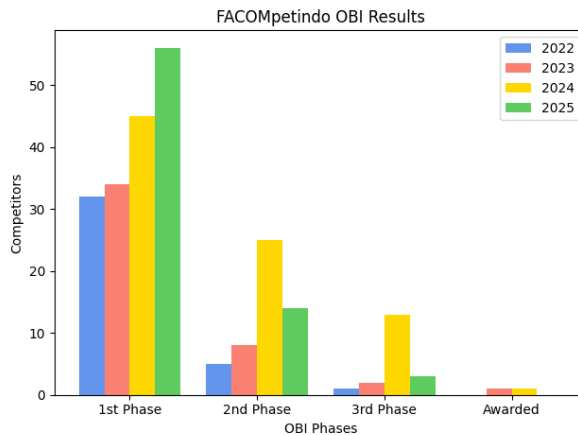


Figure 3. OBI' participation

The first phase of the MFP is conducted online, however, since 2025, the event has been held in person at FACOM for interested competitors in the region, offering an immersive environment, real-time leaderboard, snacks and prizes. That same year, FACOM had a total of 28 students registered, double the number compared to 2024, as shown in figure 5, with 12 of them competing in person.

⁸<https://codeforces.com/group/gcWlee7N7m/contest/586100/standings/groupmates>true>

The best classified student from the university in 2025 solved a total of 3 problems, finishing in 260 out of over 1000 competitors, unfortunately, in that year, we had no students classified for the final phase.

Like the MFP, the Phase Zero of the Marathon, which is traditionally held online, has been offered in-person for teams interested in this option, since 2024, thereby encouraging participation in this competition. In-person participation fosters a sense of community and belonging among students, encouraging participation, outreach, and interest [Nunes et al. 2024]. As a result, in 2024, 33 local teams (each with 3 students) registered, a 153% increase over the previous year, ranking UFMS as the 2nd university with the highest number of registered teams.

Unfortunately, in 2025, we had a reduction in the number of teams registered for Phase Zero, having about 20 teams from the university, however, all teams solved at least 1 problem, with the majority solving at least 3. In addition, a lot of teams were comprised of 3 freshmen. These facts suggest a very positive scenario for the future, showing real impacts from the project.

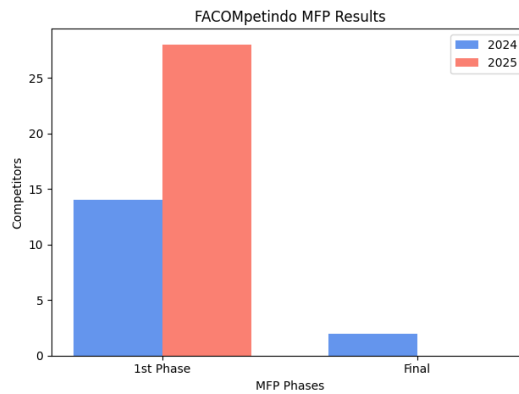


Figure 4. MFP' Participation

Also in 2025, on the First Phase of the Marathon, there were 9 teams from UFMS registered, representing an increase from 2024 and a similar amount compared to 2023. Some factors prevent this number from getting higher like in the other competitions, the biggest one is the fact that this competition has a registration fee, despite this usually being covered by the university, it isn't always. Likewise, the First Phase is disputed in teams of 3, in contrast from the OBI and MFP, this make some students not being able to participate by not knowing other interested students.

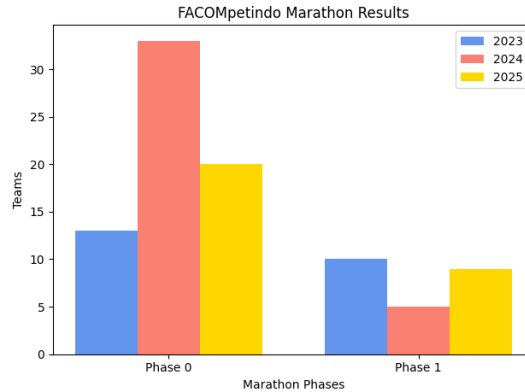


Figure 5. 0 and 1st phase' participation in the Programming Marathon

Because of the number of participating institutions being low, our region has a single slot for the Brazilian Final Phase of the Programming Marathon, consecutively, in the last six years, it has been always a team from UFMS that classifies.

4. Participants' Feedback

In total, during the trainings of the first semester of 2025, 92 different people participated in at least one day of training, with an especially high participation in the first days, that diminished in the passing of the semester, but always maintaining a base of present and interested students.

After the ending of the trainings, a feedback form was created, with various questions made to measure the impact, high and low points and the vision of the participants about the classes. A total of 15 answers were collected, representing approximately 16% of all attendees.

Of the 16 respondents, 13 are in their first semester, 1 is in their third semester, and 1 is in their fifth semester. Furthermore, 76.3% are male and 26.7% are female. Regarding their course of study, 46.7% are studying Software Engineering, 33.3% are studying Computer Science, 13.3% are studying Computer Engineering, and 6.7% are studying Information Systems.

Firstly, the participants were presented with the sentences shown in Figures 6 and 7, indicating that the suggested problems are a good complement to the in-person classes.

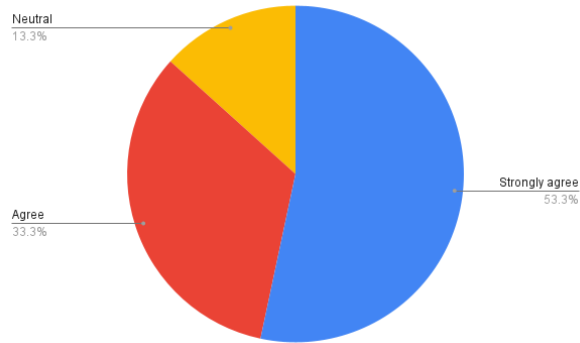


Figure 6. Sentence: "You feel more prepared to solve programming exercises due to your participation on the training."

Two sentences about the training spreadsheet were presented, as shown in Figures 8 e 9, suggesting that this resource boosts participation and strengthens learning.

The sentence "Participation in the training motivated you to participate in one or more programming competitions (MFP, OBI, Phase Zero Marathon)" received 100% "Strongly agree" responses. Additionally, when asked if they formed teams and/or studied with colleagues because of the training, 73.3% of students indicated yes, while 26.7% indicated no. In contrast, when asked about their expectations regarding the Marathon, Figure 10 shows the results, suggesting that students developed an interest in participating

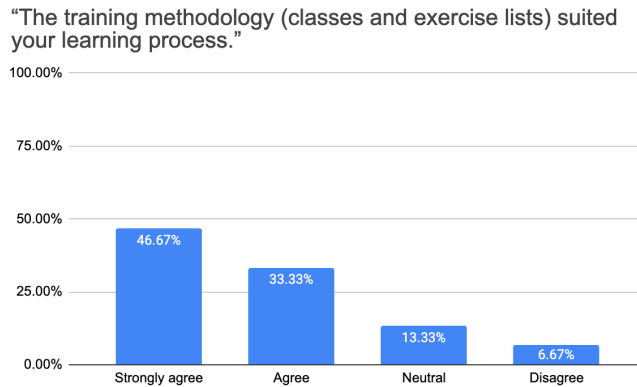


Figure 7. Sentence: "The training methodology (classes and exercise lists) suited your learning process."

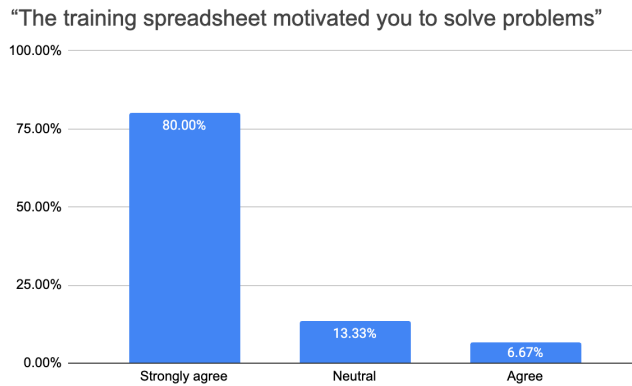


Figure 8. Sentence: “The training spreadsheet motivated you to solve problems“

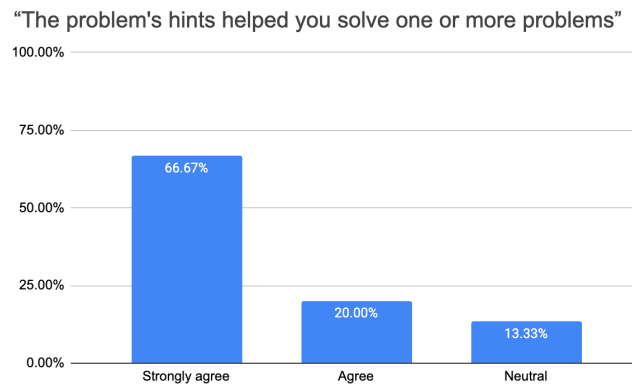


Figure 9. Sentence: “The problem’s hints helped you solve one or more problems“

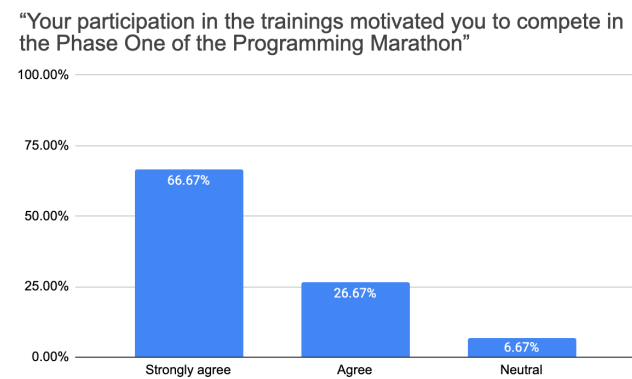


Figure 10. Sentence: “Your participation in the trainings motivated you to compete in the Phase One of the Programming Marathon“

in the competitions and many formed friendships and/or groups in this context, but the Phase One presents a greater challenge.

Both the questions "Would you like more competition simulations during training?" and "Would you recommend the training to other colleagues?" received 100% positive responses. Participants recommended the creation of additional materials, such as complete written solutions to exercises and class summaries when asked. Finally, in response to the question "What was the biggest difficulty you faced trying to participate in the training?", 6 (37.5%) students pointed to the training schedule, 4 (25%) chose the C++ programming language, and 6 (37.5%) cited the difficulty of balancing it with their university studies.

5. Conclusion

This article presented the methodology and results of the competitive programming training conducted by the FACOMpetindo group in the first semester of 2025. Overall, the participation and performance of FACOM students has never been higher, with several participation records broken and numerous students engaged. The significant participation of Software Engineering students is a new development which emerged in 2025, showing that more and more new people are being impacted. Also, the creation of supplementary materials such as spreadsheets, tips, and slides reinforces and further encourages the participation of these students. In the future, the training will continue to be implemented, considering the suggested changes.

References

- Brito, P., Fortes, R. S., Daria, F., Lopes, R., Santos, V. and Magalhães, F. (2019). Programação competitiva como ferramenta de apoio ao ensino de algoritmos e estrutura de dados para alunos de ciencia da computação. *Reserach Gate*, pages 359–368.
- Manev, K. (2024). Preparing of youngest students for participation in programming contests. *OLYMPIADS IN INFORMATICS*, pages 63–80.
- Moura A. F., Tavares T., M. J. and G., M. (2019). Virtualização de questões da obi para o desenvolvimento do pensamento computacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 8, page 1334.
- Nair, P. R. (2020). Increasing employability of indian engineering graduates through experiential learning programs and competitive programming: Case study. *Procedia Computer Science*, 172:831–837. 9th World Engineering Education Forum (WEEF 2019) Proceedings: Disruptive Engineering Education for Sustainable Development.
- Nunes, D. S. N. (2022). Programação Competitiva IFB: Módulo 01. IFB Programação Competitiva.
- Nunes, J., Escalante, L., Silva, L. and Penze, L. (2024). A primeira maratona feminina de programação do brasil: motivações para o desenvolvimento do projeto e relatos da primeira edição do evento. In *Anais do XVIII Women in Information Technology*, pages 1–11, Porto Alegre, RS, Brasil. SBC.
- Quintela, B. and Ribas, A. M. (2016). Ensinar e aprender algoritmos e programação no ensino superior: Desafios e melhores práticas. *Reserach Gate*, pages 1–16.

- R. Raman, H. V. and Achuthan, K. (2018). Students motivation for adopting programming contests: Innovation-diffusion perspective. In *Education and Information Technologies*, pages 23(5): 1919–1932.
- Ramos, G. N., Costa Jr., E. A. and Borges, V. R. P. (2025). Maratona de programação: Rumo ao futuro. *Computação Brasil*, (53):24–28.
- Silva, T., e Silva, G. B., Silva, M. and Araújo, M. (2023). Maratonando! inspirando e capacitando programadores com diversidade de gênero e variedade de competições. In *Anais do XVII Women in Information Technology*, pages 346–351, Porto Alegre, RS, Brasil. SBC.
- Verdugo-Castro, S., García-Holgado, A. and Sánchez-Gómez, M. C. (2022). The gender gap in higher STEM studies: A systematic literature review. *Heliyon*, 8(8).
- Vianna, A. C. G., Silva, M., Pegoraro, R. and Yonezawa, W. M. (2020). Programação competitiva -a união entre a matemática e computação. *PRP USP*, pages 1–6.
- Wageed, E., Elgamal, Y., Ismail, O. and Abdrabou, M. (2024). The impact of non-formal educational approach on the academic performance and employability of engineering and computer science students. *OLYMPIADS IN INFORMATICS*, pages 129–145.



Gabriel P. Falcão is a Computer Science student at UFMS (*Universidade Federal de Mato Grosso do Sul*), and has completed an academic exchange semester in Information Science at the University of Porto, Portugal. He is a founding member of FACOMpetindo, a student-led initiative focused on competitive programming training, where he contributes as a mentor and organizer. His work involves developing educational materials, guiding beginner students, and promoting participation in programming contests.

He has been actively involved in competitive programming both as a contestant and as an organizer of events, and currently serves as a member of the Problem Setting team of MFP (Maratona Feminina de Programação), an important competition in Brazil.



Jenniffer O. Checchia is a Computer Science student at UFMS (*Universidade Federal do Mato Grosso do Sul*) with research focused on Education and Gamification processes since 2020 (at *Instituto Federal do Espírito Santo*), and in Computer and Robotics Education since 2024. Currently, research AI application in educational contexts, 3d printing education and also supporting FA-COMpetindo.



Erick Ribeiro is a Computer Engineering student at UFMS (*Universidade Federal do Mato Grosso do Sul*), who is involved in competitive programming by organizing training for freshmen, creating materials and teaching classes in the FACOMpetindo project, in addition to having participated in the ICPC twice, in the ICPC Summer School at Unicamp, and having won a bronze medal at the OBI in 2019.



Mariana G. Ramires is a Computer Science student at UFMS (*Universidade Federal do Mato Grosso do Sul*). Her research focuses on Artificial Intelligence in Law and Fairness in Machine Learning. Beyond her academic work, she is an active member of the FACOMpetindo community, contributing as both a competitor and an organizer, with a dedicated interest in female and other underrepresented genders participation in competitive programming.



Matheus Amorim is a Computer Engineering student at UFMS (*Universidade Federal do Mato Grosso do Sul*). As a competitor, he participated in two National Programming Marathon Finals (ICPC). In addition to competing, he is actively involved in the development of the local community, teaching and training other students on algorithms in the FACOMpetindo project.



Amaury Antônio de Castro Junior is an Associate Professor at the Faculty of Computing at the Federal University of *Mato Grosso do Sul*, with a master's degree in Computer Science from UFMS (2003) and a doctorate from USP (2009). His research has focused on topics related to Computer Science Education, Educational Robotics, Computing in Basic Education, Programming Language Design, and the development of didactic and pedagogical materials for teaching computing.



Anderson Araujo holds a degree in Computer Science from the Federal University of *Mato Grosso do Sul* (UFMS, 2004), a master's degree in Informatics from the Federal University of Paraná (UFPR, 2006), and a doctorate in Sciences from the Technological Institute of Aeronautics (ITA, 2014). Actively worked as a researcher in the research groups: IMAGO (Computer Graphics and Image Processing -UFPR) and AIRGroup (Artificial Intelligence and Robotics Group -ITA). Was a researcher at the computational intelligence laboratory of the Federal Revenue Service and in several other research projects and private companies. Works at LIA (Artificial Intelligence Laboratory -Facom/UFMS) mainly on the following topics: Machine Learning, Computer Vision, Intelligent Agents, and Reinforcement Learning. Currently a faculty member at the computer science department of UFMS. He served as Director of Academic Planning and Management for the Pro-Rectorate of Undergraduate Studies and is the current director of the Information and Communication Technology Agency at UFMS.



Said Adi holds a degree in Computer Science from the Federal University of *Mato Grosso do Sul* (1997), a master's degree in Computer Science from the University of São Paulo (2000), and a doctorate in Computer Science from the University of São Paulo (2005). He worked as a visiting professor at the Laboratory of Biometrics and Evolutionary Biology at the University of Lyon 1, France (2018), with funding from CAPES within the context of Strategic Programs -DRI. Currently, he is a full professor at the Federal University of Mato Grosso do Sul, working in the area of Computer Science, with an emphasis on Theory of Computation (Analysis of Algorithms and Computational Complexity), Computational Biology, and, more recently, in the area of Computing and Education.

Interdisciplinarity as an Entry Point into Computer Science: Women’s Pathways, Identity, and Belonging in Hybrid Higher Education

Laura MARRONE BERZETTI DI BURONZO

Università di Modena e Reggio Emilia, Libera Università Internazionale degli Studi Sociali Guido Carli, National Ph.D. Program in Learning Sciences and Digital Technologies, Italy
e-mail: lmarrone@luiss.it

Abstract. The study examines how academic interdisciplinarity influences women’s engagement with STEM, building on literature showing that integrating Humanities and Social Sciences enhances critical thinking, ethical awareness, and inclusiveness in techno-scientific education. While prior research has mainly focused on institutional and pedagogical outcomes, less is known about women’s lived experiences of hybrid pathways: using qualitative Thematic Analysis of interviews with nine female students enrolled in hybrid STEM programs, the findings indicate that interdisciplinarity operates as a relational process, shaping women’s trajectories, perceptions of belonging, and persistence in STEM, as well as their engagement in competitive environments like Informatics Olympiads. In fact, integrated pathways make these domains more accessible, enabling the construction of new hybrid identities, and providing resources for navigating gendered environments - including highly competitive settings such as IT contests - although their effectiveness depends on institutional recognition.

Keywords: STEM Literacy, Interdisciplinarity, Gender-Inclusive Learning, Computer Science.

1. Introduction

Traditionally, techno-scientific fields have been viewed as distinct from the Humanities, grounded in the longstanding separation between empirical inquiry and the interpretative, value-laden perspectives peculiar to humanistic disciplines. This disciplinary split has reinforced the idea that STEM subjects are all about objectivity, measurement, and technical proficiency, while the Humanities’ focus is on critical reflection, cultural analysis, and ethical debates. However, a growing interdisciplinary body of research underscores the benefits of bridging these domains to cultivate higher-order thinking needed to address the societal implications of scientific and technological developments (Sánchez & Martínez, 2021; Helligsø, 2023).

In fact, the multidimensional nature of twenty-first-century global challenges requires interdisciplinary approaches to problem-solving that extend beyond purely technical expertise (Scholkmann *et al.*, 2023): in this scenario, the integration of Social Sciences and Humanities perspectives into STEM education represents a significant pedagogical shift toward fostering a more holistic understanding of complex contemporary problems. These modern issues require solutions informed not only by technical expertise but also by cultural, social, ethical, and historical perspectives (Faulconer *et al.*, 2020; Clark, 2025).

It has been demonstrated that this educational approach enhances students' problem-solving abilities, communication skills, and preparedness for addressing global phenomena such as climate change and infectious diseases, complex issues that demand analyses extending well beyond conventional disciplinary boundaries (Joseph & Uzundu, 2024; Park & Cho, 2022). By bridging ethical, cultural, and social considerations into STEM curricula, such approaches not only deepen students' understanding of scientific interconnectedness but also strengthen their capacity for responsible decision-making in navigating socio-scientific challenges (Mudaly & Chirikure, 2023).

2. Related Work

Even though traditional STEM education provides a strong foundation in scientific and technological skills, it often falls short in cultivating the critical thinking and civic awareness required to address the long-term consequences of global changes (Lee *et al.*, 2023). An example of this is the tendency of traditional engineering pedagogy to prioritize technical efficiency over environmental, ethical, and social considerations (López-Santiago *et al.*, 2024; Cañavate *et al.*, 2025).

Hybrid educational frameworks thus diverge from traditional STEM pedagogies that prioritize objectivity and procedural knowledge while neglecting socio-cultural and philosophical concerns (Singh, 2020): by embedding moral and ethical deliberation within scientific learning and foregrounding real-world contexts, complex problems are analysed through the interplay of scientific evidence and human values (Ramos & Mendonça, 2021; Zeidler & Karişan, 2024; Roehrig *et al.*, 2021). This encourages a critical evaluation of both scientific information and its societal implications (Johnson *et al.*, 2024; Ke *et al.*, 2023), deepening students' understanding of science while enhancing their awareness of its societal dimensions (Viehmann *et al.*, 2024).

In this context, Socio-Scientific Issues (SSI) – i.e., socially relevant, controversial, real-world problems that are informed by science and often include an ethical component (Sadler *et al.*, 2007) - represent a powerful educational tool for demonstrating the social relevance of STEM subjects and therefore strengthening scientific consciousness and active citizenship (Alcaraz-Domínguez & Barajas, 2021). As a matter of fact, the contested viewpoints and the absence of clear solutions that characterize these issues imply that they cannot be resolved solely through scientific knowledge: the latter has to be blended with ethical, political, and economic considerations in order to face complex scenarios that lack straightforward answers (Lin *et al.*, 2018; Zeidler & Karişan, 2024; Högström *et al.*, 2024; Garrecht & Adler, 2025).

By requiring students to draw on evidence and to assess competing moral considerations, real-world dilemmas become vehicles for meaningful learning that highlight the social embeddedness of science, enhancing students' grasp of the nature of scientific inquiry (Gutierrez, 2014; Güler, 2023; Johnson *et al.*, 2024; Kinskey & Newton, 2024). In fact, integrated education – by compelling students to confront their preconceived notions, refine their interpretations through dialogue and reflection, and integrate both scientific and moral considerations into their reasoning (Karahana, 2022; Hanifha *et al.*, 2023; Zeidler *et al.*, 2019) - not only helps them to situate scientific progress within broader social narratives, accounting for its impacts on communities (Hwang *et al.*, 2023; Lee *et al.*, 2023), but also helps them to connect scientific knowledge to everyday life. By recognizing how scientific developments intersect with ethical, social, and environmental concerns, learners develop robust critical thinking and problem-solving skills (Helligsø, 2023; Karışan & Zeidler, 2024; Bezen, 2024) that allow them to evaluate the moral nuances of contemporary technological societies (Martorell *et al.*, 2024; Turgut & Yakar, 2021; Huong *et al.*, 2024; White & Delaney, 2021).

Socio-scientific learning therefore fosters the “functional scientific literacy” (Tabak, 2016) necessary for informed decision-making in contexts where ethical and scientific dimensions are deeply intertwined (Widiyanti & Susilayati, 2023; Sjöblom *et al.*, 2023; Zeidler *et al.*, 2019; Johnson *et al.*, 2020; Lee *et al.*, 2023; Burkett, 2025). This requires a complex blend of capabilities beyond traditional disciplinary boundaries (Wajngurt & Sloan, 2019). Integrating the arts, humanities, and social sciences helps cultivate “humanistic engineers” with civic responsibility and the capacity to critically engage with the intellectual foundations of science (Lambrinidou & Edwards, 2020), leading to moral development and community engagement (Chowdhury *et al.*, 2020), and thus to a comprehensive understanding and reflective judgment (Özbek & Aslan, 2024). This integration involves more than simply adding “soft skills” to technical courses: it requires a deep transformation of the underlying value systems of higher education (Helligsø, 2023) by reframing technical problems through ethical and societal lenses (Berg & Lee, 2016). This ultimately encourages students to adopt broader perspectives (Coxon, 2021): by examining how risks and benefits are distributed and how technical and scientific practices may reinforce or mitigate inequities (Verma, 2024), students learn to identify technologies' implicit values and to provide solutions that are both technically correct and socially just (Forbes & Hoople, 2023).

Thanks to this dynamic re-configuration of STEM education as a process that integrates ethical reflection, social justice, and community engagement, it is possible to co-create more inclusive and socially responsible solutions (Lucena & Leydens, 2015) by asking questions like “Who benefits from this technology?” and “What constitutes a just outcome?”. This new approach to science promotes the interrogation of power dynamics, the examination of diverse perspectives, and the prioritization of equity in technological development, putting human values at the centre of innovation (Imad *et al.*, 2023; Lim *et al.*, 2021; Nieusma, 2020). This transformative approach prepares students to be not only scientifically competent, but also ethically and culturally aware of the consequences of their actions (Gutierrez, 2014; Leij *et al.*, 2021; Bezen, 2024; Kirby, 2020): this allows

them to be prepared to participate in public debates around STEM-related issues (Alcaraz-Domínguez & Barajas, 2021), responding to longstanding problems in STEM education, where ethical and social dimensions have often been marginalized (Beatty *et al.*, 2023). In particular, integrating STEM with the humanities fosters a deeper understanding of cultural, social, and historical contexts (Clark, 2025), encouraging the use of technical knowledge to address social inequities (Lee *et al.*, 2023).

This hybridization is also fundamental in rapidly evolving domains like AI, enabling students to explore ethical dilemmas in immersive environments and, when combined with mathematical modelling and multidisciplinary inquiry, strengthening interdisciplinary education and enriching learning by providing personalized feedback on students' ethical judgment (Clark, 2025). In fact, AI-based tools are able to support collaboration and critical reflection on the ethical challenges inherent to STEM disciplines (Børsen & Butkevičienė, 2023; Chen *et al.*, 2025), and case studies involving AI applications, of course, improve both AI literacy and ethical awareness (Usher & Barak, 2024). In this way, students are prepared to navigate the multifaceted implications of technological progress, fostering informed and responsible innovation (Chen *et al.*, 2025) in fields where students have traditionally relied on unstructured moral intuition (Børsen & Butkevičienė, 2023). For example, Virtual Reality tools create immersive settings for ethical decision-making (Tobías *et al.*, 2025), while adaptive tutoring systems personalize these experiences and reinforce ethical reasoning (León *et al.*, 2025; Clark, 2025). AI-powered virtual laboratories enable simulated experiments while addressing real-world ethical implications, strengthening both scientific understanding and moral deliberation (Joseph & Uzundu, 2024; See *et al.*, 2023).

By incorporating elements from the social sciences and humanities, interdisciplinary STEM pedagogies effectively demonstrate how diverse academic interests can converge, offering a more holistic learning experience. This expanded pedagogical vision resonates with a diverse student body, particularly women, who often seek broader societal relevance in their STEM trajectories. In fact, evidence suggests that interdisciplinary curricula make STEM fields more accessible and better aligned with female students' personal values and aspirations (Joseph & Uzundu, 2024). By recognizing the broader range of interests and capabilities among female students (Helligsø, 2023), female retention rates are enhanced.

In higher education, the shift toward holistic STEM education is represented by interdisciplinary academic programs (Chen *et al.*, 2023; Hitt *et al.*, 2023), which stress the interconnectedness of scientific knowledge, human values, and societal well-being (Chen & Hoople, 2018; Niles *et al.*, 2020; Chen *et al.*, 2023) by bridging disciplinary divides. Globally, universities are increasingly adopting hybrid STEM programs to address persistent gender gaps in STEM. Pedagogically, these programs often utilize project-based learning, inquiry-driven methods, and teamwork, approaches strongly associated with higher engagement and retention among underrepresented groups (Costa *et al.*, 2023). Successful initiatives such as the Technical University Dublin (TUD) model - based on empirical research that demonstrates that females do not lack skills but have a diverse profile of abilities (Ovalle Ramirez, 2025) - demonstrate that hybrid STEM programs are effective

in attracting and retaining women by recognizing broader interests, consequently reducing their perceived risk and increasing their perceived self-efficacy and outcome expectations (Aizenman *et al.*, 2022). However, scientific evidence indicates that interdisciplinarity alone is insufficient without institutional enablers, such as mentorship structures, internships, faculty training, interdepartmental coordination, scholarships, tailored program orientation, and sustained academic monitoring. In particular, academic literature identifies a set of consistent principles capable of stimulating girl's interest in STEM, and consequently counteracting the gender gap that still characterizes these subjects: the integration of "explicit relevance" through real-world problem-solving; the use of active learning and collaborative practices; strong mentorship networks; robust institutional support and long-term monitoring (Freeman *et al.*, 2014).

From a psychological point of view, the key-mechanisms involved in these virtuous processes include enhanced Utility Value, where perceived usefulness and relevance boost other aspects of a person's life, like overall performance and persistence (Wigfield, 1994; Peterson, 2000; Emerick, 1992), and social identity, which is particularly important given that early sense of STEM-belonging predicts achievement, well-being, and resilience (Rainey *et al.*, 2018). In this regard, hybrid curricula provide alternative points of entry (Watkins-Lewis *et al.*, 2022), expanding the range of identities women can envision for themselves in science and consequently reducing stereotype threat (Potter, Crispin, Dollard *et al.*, 2025).

In conclusion, a growing body of research shows that interdisciplinary programs can effectively broaden the appeal of STEM, promoting a more inclusive understanding of scientific practice (Cobian *et al.*, 2024; Türkkan *et al.*, 2024): the fact that STEM domains are not seen as merely technical anymore, but as disciplines with a strong social relevance, is a pedagogical shift able to attract a more diverse student population (Romkey, 2020). By connecting scientific learning to personal and structural issues (Lee *et al.*, 2023), hybrid frameworks mitigate the feelings of alienation commonly experienced by women and minorities in conventional STEM environments (Valla & Ceci, 2014; Watkins-Lewis *et al.*, 2023): this aligns scientific pursuits with prosocial goals, which research shows are particularly salient for women and strongly influence their decisions to pursue techno-scientific careers (Miller & Withers, 2023).

However, while this outcome-oriented perspective highlights the pedagogical, ethical, and civic benefits of integrated STEM education, less attention has been paid to how interdisciplinarity operates at an individual level, particularly for women. Indeed, what remains largely underexplored is how women subjectively experience interdisciplinarity in their personal and academic STEM journeys. In particular, existing studies rarely examine interdisciplinarity as a relational and experiential mechanism through which women negotiate their positioning within STEM fields, rather than as a pedagogical feature or institutional intervention alone.

3. Methodology & Research Design

The present study aims to explore how interdisciplinarity shapes women's subjective pathways into and through STEM, i.e., how they live and interpret hybrid pathways over time. To do so, interdisciplinarity is not conceptualized as an abstract curricular principle, but rather as a structuring mechanism that reshapes women's relationships to STEM disciplines, identities, and institutional contexts.

Given the explorative nature of the research question and the strong focus on meaning-making, sense of identity, and lived experience, the study complements existing quantitative and program-evaluation research with a processual and relational perspective, choosing a qualitative design grounded in Thematic Analysis (Braun & Clarke, 2006; Braun & Clarke, 2021). This allows the study to build directly on prior work on inclusive STEM education while advancing a deeper understanding of how interdisciplinarity can function as a device of access, legitimacy, and persistence for women in STEM, mitigating gendered individual experiences of exclusion in male-dominated environments.

Semi-structured, in-depth interviews have been conducted with 9 female participants enrolled in interdisciplinary STEM-related courses at the Roman universities of Tor Vergata and Luiss. Interviews focused on participants' educational journeys, motivations, perceptions of interdisciplinarity, experiences within STEM environments, and reflections on gender, belonging, and future aspirations. This format allowed for both comparability across interviews and flexibility to follow issues that participants themselves identified as significant.

Data have been then analysed following the six-phase approach required by Thematic Analysis. The process has been inductive, without imposing predefined categories, in order to remain close to participants' accounts. It involved:

1. Familiarisation with the data through repeated reading of transcripts;
2. Initial coding focused on meaningful segments related to interdisciplinarity, STEM engagement, and identity;
3. Development of focused codes capturing recurring patterns;
4. Construction and refinement of themes through iterative comparison;
5. Review and naming of themes to ensure internal coherence and analytical distinction;
6. Production of thematic narratives linking empirical patterns to the research questions.

Analytic dimensions included the role of interdisciplinarity as an entry point, identity resource, and buffer against gendered exclusion. Analytical rigor was ensured through iterative coding, systematic memo writing, and ongoing comparison between cases. Reflexivity was maintained throughout the research process, with attention to the researcher's positionality and interpretive role.

4. Results

Three overarching themes were identified regarding how academic hybridization shapes women's engagement with STEM, capturing the role of interdisciplinary configurations as entry points, identity resources, and mechanisms for navigating gendered STEM environments.

4.1 *Theme 1: Interdisciplinarity as a legitimate entry-point into STEM.*

Across interviews, interdisciplinarity emerged as a key mechanism enabling women's initial engagement with STEM. Rather than entering STEM through narrowly defined technical pathways, many participants encountered STEM through hybrid domains that integrated technical knowledge with social, humanistic, or applied perspectives. This interdisciplinary framing reduced the perceived distance from STEM and made technical content more accessible and meaningful. Participants often described interdisciplinary programmes as providing a "safe" or legitimate space from which to approach technical disciplines without requiring early identification with traditional STEM stereotypes. As such, interdisciplinarity served as a gateway, broadening access to STEM beyond conventional entry routes.

4.2 *Theme 2: Hybrid Identities and negotiated STEM-belonging.*

Interdisciplinarity also played a central role in shaping participants' identities and sense of belonging within STEM. Rather than adopting singular disciplinary identities, participants articulated hybrid self-understandings that combined STEM competencies with skills and values from other fields, such as communication, ethics, creativity, and social impact. These hybrid identities enabled participants to engage with STEM while resisting narrow or exclusionary definitions of what it means to be "good at STEM". In doing so, interdisciplinarity supported alternative modes of belonging that were perceived as more aligned with participants' interests and self-conceptions. This finding highlights how interdisciplinary positioning can expand the symbolic boundaries of STEM participation.

4.3 *Theme 3: Interdisciplinarity as a buffer against gendered exclusion.*

A third theme concerns the role of interdisciplinarity in mitigating gendered exclusionary experiences in STEM environments. Participants frequently described male-dominated contexts in which women's competence or legitimacy was implicitly questioned. In such settings, an interdisciplinary positioning provided both symbolic and practical resources for navigating exclusion. By occupying roles at the intersection of disciplines, participants were able to assert value through mediation, translation, or applied problem-solving, rather than through direct competition on narrowly technical criteria. However, the effectiveness of this buffering function depended on institutional recognition. Where interdisciplinary programmes were structurally legitimised, participants reported greater stability and confidence; where hybridity was marginalised, its protective effect was weaker.

5. Implications for Informatics Olympiads

The study's findings offer particularly relevant insights when applied to the context of Informatics competitions, which nowadays represent one of the most prominent environments for training and recognizing concrete technical talent within STEM.

In fact, contests are often structured around a strongly disciplinary and technical conception of the discipline, centred on algorithmic problem-solving, optimization, and individual performance. However, in light of the findings presented in this research, such a model risks reproducing some of the dynamics of exclusion and limited belonging that participants describe in more traditional STEM environments.

In particular, the first theme - interdisciplinarity as an entry point - suggests that access to technical domains can be significantly broadened when they are presented alongside social, applied, or humanistic dimensions. Informatics Olympiads, by contrast, tend to privilege early and highly selective forms of access that presuppose strong identification with an abstract, competitive paradigm of the discipline. This may constitute an implicit barrier for many female students, who, as the study shows, often develop an interest in STEM through hybrid, contextualized pathways rather than through purely technical exposure. From this perspective, one key-implication concerns the reconfiguration of entry and preparation pathways for Olympiad participation. Integrating interdisciplinary elements - such as problems embedded in social, ethical, or real-world contexts - could increase the perceived relevance and accessibility of computer science, enhancing what the literature defines as "utility value" (Wigfield, 1994; Hulleman *et al.*, 2010; Peterson, 2000). Importantly, this would not entail reducing technical rigor, but rather reframing it within concrete contexts, in line with the findings that perceived relevance plays a crucial role in female engagement.

The second theme, concerning hybrid identities and negotiated belonging, is equally significant. Informatics Olympiads tend to construct and reward a form of "excellence" often associated with speed, individual performance, and mastery of advanced technical techniques. However, the findings indicate that many women develop a sense of belonging in STEM through more composite identities that include communication skills, ethical awareness, interest in social impact, and the ability to bridge different domains. This suggests that current Olympiad formats may fail to fully recognize alternative forms of excellence. One possible direction for development could involve introducing parallel or complementary formats to traditional competitions, such as interdisciplinary hackathons, collaborative challenges, or problem sets that explicitly require integrating computer science with other domains (e.g., sustainability, public policy, or AI ethics). Such formats could legitimize a plurality of ways of "being good at computer science," thereby expanding the symbolic boundaries of participation.

The third theme - interdisciplinarity as a buffer against gendered exclusion - offers perhaps the most critical implication. Informatics Olympiads are often highly competitive and male-dominated environments, where dynamics similar to those described by participants may emerge: the need to constantly prove one's legitimacy, experiences of isola-

tion, and evaluation based on narrow criteria of competence. In such contexts, the absence of interdisciplinary dimensions may limit the symbolic and practical resources available to navigate these challenges. By contrast, incorporating interdisciplinary elements could provide additional modes of participation and recognition, enabling female students to position themselves not only as technical competitors but also as interpreters, mediators, and socially oriented problem solvers. However, consistent with this study's findings, the effectiveness of such strategies would depend strongly on institutional recognition: if interdisciplinary components are perceived as marginal or secondary to the "core" competition, their impact on belonging and inclusion is likely to remain limited.

Another important aspect concerns the role of Olympiads as early orientation devices. These competitions contribute to defining who is seen as "belonging" to computer science from an early stage, shaping academic choices and future trajectories. If, as this study suggests, interdisciplinarity expands the range of possible identifications with STEM, then a strictly disciplinary Olympiad system may inadvertently restrict these possibilities, selecting not only based on ability but also on alignment with a specific identity model.

Finally, these findings invite reflection on the fact that Informatics Olympiads are not merely mechanisms for talent selection, but also cultural environments that implicitly convey values, norms, and representations of the discipline. In this sense, even partial integration of interdisciplinary perspectives could promote a vision of computer science not only as a technical field but also as a situated, socially relevant practice open to diverse forms of participation.

In conclusion, applying an interdisciplinary lens to Informatics Olympiads does not undermine their competitive nature or technical rigor, but rather enriches their educational and inclusive potential. In line with this study's findings, this could represent a key lever for broadening access, strengthening belonging, and mitigating exclusionary dynamics, ultimately contributing to more equitable and representative pathways into STEM.

References

- Aizenman, A., Roberts, L., & Chen, M. (2022). Interdisciplinary research and STEM-focused social science curriculum support retention, STEM identity, and outcome expectations. *Journal of STEM Education Research*, 5(2), 145–168.
- Alcaraz-Domínguez, S., & Barajas, M. (2021). Socioscientific issues and scientific citizenship: Toward informed participation. *Science Education Review*, 32(1), 22–38.
- Beatty, R., Lee, J., & Enders, K. (2023). Ethical gaps in STEM curricula: A systematic review. *Journal of STEM Ethics*, 4(1), 1–19.
- Berg, M., & Lee, C. (2016). Reframing engineering ethics through societal lenses. *Engineering Education Quarterly*, 17(3), 201–223.
- Bezen, S. (2024). Connecting science to everyday life: A socioscientific approach. *International Journal of Science Education*, 46(2), 155–172.
- Biemmi, I. (2015). Gender in schools and culture: Taking stock of education in Italy. *Gender and Education*, 27(1), 1–16. <https://doi.org/10.1080/09540253.2015.1103841>

- Børsen, T., & Butkevičienė, E. (2023). Ethics in AI education: Challenges and opportunities. *AI & Society*, 38(1), 55–72.
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101. <https://doi.org/10.1191/1478088706qp063oa>
- Braun, V., & Clarke, V. (2021). *Thematic analysis: A practical guide*. SAGE.
- Cañavate, J., Romero, P., & Díaz, L. (2025). Rethinking engineering pedagogy: Integrating environmental and ethical concerns. *European Journal of Engineering Education*, 50(1), 44–63.
- Chen, A., & Hoople, G. (2018). Humanistic engineering: Integrating values into technical education. *Engineering Studies*, 10(2), 95–112.
- Chen, A., Hoople, G., & Rivera, S. (2023). Humanities-integrated engineering: New directions in interdisciplinary training. *Journal of Engineering Education*, 112(3), 589–612.
- Chen, M., Patel, R., & Lin, H. (2025). AI-supported ethical learning in engineering. *Computers & Education*, 214, 104949.
- Chowdhury, M., Hasan, S., & Karim, F. (2020). Socioscientific issues in science education: Moral development and community engagement. *Journal of Moral Education*, 49(4), 567–584.
- Faulconer, E., Ely, M., & McLaughlin, J. (2020). Integrating humanities into STEM for 21st-century competencies. *Journal of Interdisciplinary Education*, 12(3), 245–263.
- Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410–8415.
- Güler, G. (2023). Socioscientific reasoning in secondary science classrooms. *Journal of Research in Science Teaching*, 60(7), 1123–1145.
- Gutierrez, K. (2014). Learning through real-world dilemmas: A sociocultural approach. *Educational Researcher*, 43(4), 235–248.
- Hamad, L., Ortiz, F., & Singhal, P. (2024). Responsible innovation in engineering education. *Technology in Society*, 56, 101–119.
- Helligsø, S. (2023). Humanities integration in STEM: Educational reform for a complex world. *Scandinavian Journal of Educational Research*, 67(4), 554–573.
- Hitt, K., Douglas, E., & Walters, K. (2023). Interdisciplinary engineering pathways in higher education. *International Journal of Engineering Pedagogy*, 13(1), 74–95.
- Högström, M., Pettersson, P., & Norén, M. (2024). Ethical deliberation in science classrooms. *Journal of Science Education and Technology*, 33(2), 198–215.
- Hulleman, C. S., Godes, O., Hendricks, B. L., & Harackiewicz, J. M. (2010). Enhancing interest and performance with a utility value intervention. *Journal of Educational Psychology*, 102(4), 880–895. <https://doi.org/10.1037/a0019506>
- Johnson, B., Harris, L., & Moon, T. (2020). Functional scientific literacy in complex societies. *Journal of Science Literacy*, 11(1), 1–19.
- Johnson, T., Rivera, A., & López, M. (2024). Embedding ethics in STEM through socioscientific learning. *Journal of STEM Pedagogy*, 6(2), 113–132.
- Joseph, P., & Uzundu, C. (2024). Global challenges and interdisciplinary STEM learning. *International Journal of STEM Education*, 11(2), 88–110.
- Karahan, E. (2022). Socioscientific discussions and conceptual change. *Eurasian Journal of Science Education*, 14(3), 345–360.

- Karışan, D., & Zeidler, D. L. (2024). Moral reasoning in socioscientific contexts: New directions. *International Journal of Science Education*, 46(5), 789–812.
- Ke, L., Sun, Y., & Wu, J. (2023). Socioscientific engagement for scientific literacy. *Journal of Science Communication*, 22(4), 1–15.
- Kirby, L. (2020). Ethics and science literacy. *Journal of STEM Literacy*, 15(1), 45–59.
- Kinskey, R., & Newton, M. (2024). Critical thinking through socioscientific controversies. *Journal of Secondary Science Teaching*, 39(1), 77–96.
- Lambrinidou, Y., & Edwards, M. (2020). Community-based engineering ethics. *Engineering Studies*, 12(1), 1–23.
- Leij, A., Mortensen, T., & Drumm, S. (2021). The civic turn in science education. *Science Education for Democracy*, 13(2), 112–130.
- Lim, D., Chan, H., & Brooks, F. (2021). Ethics and equity in global engineering. *International Journal of Engineering Ethics*, 8(2), 201–226.
- Lin, T., Chen, H., & Li, P. (2018). Navigating complex problems: Ethical decision-making in STEM. *Journal of Science and Engineering Education*, 10(4), 255–278.
- Lucena, J., & Leydens, J. A. (2015). *Engineering justice: Transforming engineering education and practice*. IEEE Press.
- Martorell, J., Campos, G., & Ruiz, M. (2024). Ethical awareness in technological societies. *Ethics & Education*, 19(1), 65–85.
- Means, B., Wang, H., Wei, X., Lynch, S., Peters, V., Young, V., & Allen, C. (2017). Expanding STEM opportunities through inclusive STEM-focused high schools. *Science Education*, 101(5), 681–715. <https://doi.org/10.1002/scs.21281>
- Miller, M., & Withers, M. (2023). Prosocial motivations and women in STEM. *Gender and Education*, 35(7), 845–865.
- Mudaly, R., & Chirikure, S. (2023). Contextualizing science learning through social justice. *Journal of Science Education Research*, 9(3), 201–223.
- Nieusma, D. (2020). Rethinking engineering for humanity. *Engineering Studies*, 12(3), 230–249.
- Niles, S., Contreras, S., Roudbari, S., Kaminsky, J., & Harrison, J. (2020). Resisting and assisting engagement with public welfare in engineering education. *Journal of Engineering Education*, 109(3), 491–507.
- Ortiz-Revilla, J. (2020). Science education and epistemological awareness. *Journal of Science Education*, 29(4), 433–450.
- Rainey, K., Dancy, M., Mickelson, R., Stearns, E., & Moller, S. (2018). Race and gender differences in how sense of belonging influences decisions to major in STEM. *International Journal of STEM Education*, 5, Article 10. <https://doi.org/10.1186/s40594-018-0115-6>
- Ramos, F., & Mendonça, P. (2021). Integrating ethics into science learning through socioscientific issues. *Journal of Science Education*, 43(2), 200–220.
- Roehrig, G. H., Dare, E. A., Ring-Whalen, E., & Wieselmann, J. R. (2021). Understanding coherence and integration in integrated STEM curriculum. *International Journal of STEM Education*, 8(1), 1–21.
- Sadler, T. D., Barab, S. A., & Scott, B. (2007). What do students gain by engaging in socioscientific inquiry? *Research in Science Education*, 37(4), 371–391.
- Scholkmann, A., Weber, L., & Reuter, M. (2023). Interdisciplinarity for global problem-solving.

- Higher Education Review, 55(3), 302–325.
- See, J., Nguyen, T., & Lim, H. (2023). AI virtual labs for ethical scientific inquiry. *Computers & Education*, 195, 104673.
- Sjöblom, A., Eriksen, T., & Lunde, K. (2023). Scientific literacy in socioscientific contexts. *Science Education International*, 34(3), 205–221.
- Tabak, I. (2016). Functional scientific literacy: Seeing the science within the words and across the web. In L. Corno & E. M. Anderman (Eds.), *Handbook of educational psychology* (3rd ed., pp. 269–280). Routledge.
- Wajngurt, C., & Sloan, E. (2019). STEM education and social responsibility. *Journal of STEM Education*, 20(2), 34–45.
- Watkins-Lewis, K., Delgado, M., & Reed, T. (2022). Belonging and access in STEM. *Journal of Diversity in Education*, 9(2), 80–104.
- Watkins-Lewis, K., Reed, T., & Carter, H. (2023). Women in STEM: Barriers and pathways. *Gender and STEM Review*, 3(1), 110–134.
- White, M., & Delaney, S. (2021). Holistic STEM education for the 21st century. *Journal of Integrated Pedagogy*, 8(3), 223–240.
- Widiyanti, R., & Susilayati, E. (2023). Functional scientific literacy and socioscientific reasoning. *Journal of Science Learning*, 12(1), 55–72.
- Wigfield, A. (1994). Expectancy-value theory in education. *Educational Psychology Review*, 6(1), 49–78.
- Younas, A., & Zeng, H. (2024). Eastern ethical traditions in STEM education. *Asian Journal of Ethics in Technology*, 3(1), 44–64.
- Zhan, Z., & Niu, H. (2023). Ethical competencies in emerging engineering programs. *International Journal of Engineering Education*, 39(6), 1125–1143.
- Zeidler, D. L., Berkowitz, M. W., & Simmons, M. L. (2019). Socioscientific reasoning and moral development. *Journal of Science Education and Technology*, 28(5), 441–453.
- Zeidler, D. L., & Karışan, D. (2024). Scientific reasoning and moral complexity. *Science Education Review*, 32(1), 55–71.



Laura Marrone Berzetti di Buronzo is a Marketing graduate specializing in Analytics and Metrics, and is currently a Ph.D. candidate in Learning Sciences and Digital Technologies at the Università di Modena e Reggio Emilia, Italy. She is deeply interested in building educational community engagement through creativity and inclusive communication, with a strong research focus on Behavioral Economics and Educational Equity. Accordingly, her diverse work experience spans from teaching Digital Marketing and serving as a Research and Teaching Assistant in Consumer Behavior to working as an Academic Orientation Adviser and collaborating on several projects related to social and educational inclusion.

Research on the Effectiveness of GPT-4o and GPT-5.4 Models in Solving Olympiad Programming Tasks on the Eolymp Platform

Andrii MYKHALKO¹, Oleksandr MITSA², Yurii HOROSHKO³, Hanna TSYBKO⁴, Viktor SHAKOTKO⁵

¹Faculty of Information Technologies, Uzhhorod National University. Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava.

²Department of Information Management Systems and Technologies, Uzhhorod National University.

^{3,4}Department of Computer Science and Computer Engineering, Taras Shevchenko National University “Chernihiv Colehium”.

⁵Department of Technological and Professional Education, Oleksandr Dovzhenko Hlukhiv National Pedagogical University.

e-mail: ¹likespro.eth@gmail.com, ²alex.mitsa@gmail.com, ³horoshko_y@ukr.net, ⁴a.tsb@ukr.net, ⁵vv0304@gmail.com

Abstract. The object of the research is the process of solving Olympiad programming tasks with Ukrainian statements from the Eolymp platform using the large language models GPT-4o and GPT-5.4.

As a result we discovered that LLM average efficiency in solving sports programming tasks with Ukrainian statements was nearly identical for C++ and Python, averaging around 32% for GPT-4o and 56% for GPT-5.4. It was revealed that LLMs effectiveness when providing Ukrainian and English statements for the same tasks was approximately the same ($p > 0.05$).

It was found that the proportion of tasks completely solved by LLMs was about 13% for GPT-4o and 37 % for GPT-5.4 of the total number considered in this study. The result for GPT-4o is comparable with the results of solving similar tasks with English statements on the Codeforces platform, while the result for GPT-5.4 is significantly larger, however, the number of tokens used by the new model GPT-5.4 is almost 3 times larger.

Keywords: Olympiad programming, large language model, GPT-4o, GPT-5.4, Eolymp, Python, C++.

1. Introduction

In the past few years, artificial intelligence (AI) has radically changed the situation in many areas, from learning (Albahijan *et al.*, 2025; Yashina, 2024) and prediction (Kotsovsky & Batyuk, 2024; Kotsovsky, 2025) to solving medical problems (Mykhalko *et al.*, 2024; Mykhalko *et al.*, 2023) and classification (Mitsa *et al.*, 2025; Kotsovsky, 2024). In many cases, it performs as well as humans, or even better, and this has already raised concerns about the possible replacement of ordinary workers by AI (AI study, 2023). One such area where some companies are already laying off people in favor of Large Language Models (LLMs) is programming. So far, however, AI tools are not “smart” enough to fully delegate most tasks in this area. Nevertheless, in the hands of an experienced programmer, LLMs can be a great helper (Dakhel *et al.*, 2023; Ferdiana, 2024).

As neural networks are currently developing at an extremely high rate (Geche *et al.*, 2022), there exist many studies that measure the effectiveness of new models in programming subfields using various metrics and methods. For example, it has been found that OpenAI’s LLM GPT-4 is extremely successful in solving problems from the LeetCode platform (Coello *et al.*, 2023) and from the Most Basic Python Problems (MBPP) set from Google researchers (Yang *et al.*, 2025). Also, in the field of intelligent code completion, it has been studied that while older AI tools mainly rely on previously known patterns, newer ones can generalize and synthesize existing knowledge into new, more complex constructions of completed code (Hou & Ji, 2024). However, the question of the effectiveness of the GPT-5.4 model in this aspect remains unexplored.

2. Analysis of literary data and problem statement

In (Jain *et al.*, 2025), the problem of comparing the performance of large language models (LLMs) with human programmers in solving algorithmic problems was considered. The authors set the goal to find out to what extent modern LLMs (GPT-4, GPT-3.5, Gemini, Claude, Llama, etc.) are able to compete with participants in online programming platforms, as well as to determine the impact of different hinting strategies and programming languages on the quality of the results obtained.

The research (Jain *et al.*, 2025) was based on experiments in the LeetCode and Geeks-forGeeks environments, where the tests covered tasks of varying complexity levels. Several interaction strategies with the models were tested: from one-time queries to iterative schemes with code execution and verification. The solutions generated by LLM were compared with the results of thousands of participants of these platforms.

The results showed that GPT-4 significantly outperformed other models and in many cases reached or even exceeded the level of most human programmers. In particular, on both platforms, GPT-4 outperformed more than 85% of participants on average and was in the top 10% in about 70% of cases. The use of iterative schemes provided the largest performance gains. In addition, GPT-4 demonstrated the ability to transform solutions between different programming languages (Python, Java, C++, JavaScript), while maintaining or even improving their correctness. The performance of the generated programs in terms of

speed and memory consumption was comparable to human solutions.

It should be noted that the study (Jain *et al.*, 2025) has limitations: the tasks used do not fully reflect the complexity of Olympiad and professional programming; the effectiveness of LLM largely depends on the correct formulation of prompts; tasks that require extensive design or deep algorithmic thinking were not considered.

In (Görmez *et al.*, 2024), the task was to investigate the development of ChatGPT models in programming, comparing their performance in solving algorithmic problems of varying complexity. Particular attention was paid to the efficiency in terms of runtime and memory usage, as well as how the choice of programming language (Python, Java, C++) affects the results.

There were selected 15 tasks from LeetCode, which were classified by complexity. The GPT-3.5, GPT-4, GPT-4o models were tested. Each task was solved in three languages: Python, Java, C++ . The code generated by the models was executed 10 times for each task, and the execution time and memory consumption were measured. Then, statistical analysis was performed, namely, two-way ANOVA and Tukey's HSD test to compare the results between models and languages. The following results were obtained.

The programming language has a significant impact on execution time and memory usage: C++ performed best among the three languages; Python was the slowest and most resource-intensive. Java is of middle level. As for comparing the GPT-3.5, GPT-4, and GPT-4o models, no statistically significant differences in performance were found for most tasks. The models cope successfully with the tasks of medium and low complexity, but performance dropped significantly at high complexity. Although ChatGPT models show great potential in helping with algorithmic tasks, their practical utility is limited in cases where resources (time, memory) or high level of complexity are important. The choice of programming language is important: using C++ can significantly improve performance if resource constraints matter.

Therefore, the authors emphasize that ChatGPT is better viewed not as a replacement for a human programmer, but as a complementary tool, especially for tasks of medium complexity, refactoring, or for getting help with code.

In (Coello *et al.*, 2024) the following questions were considered:

1. What LLMs are used in software engineering tasks - their architecture, properties, areas of application.
2. How are data / datasets used for such models collected and processed - quality, purity, data preparation.
3. What are the optimization strategies and methods for evaluating the effectiveness of models in software engineering tasks.
4. For which specific tasks did the studied models give the best or noticeable results; where have they been successfully applied.

The studied models include both general-purpose large language models (LLMs) (e.g., GPT-like, Codex, CodeGen), and models specialized in code.

Regarding datasets, it is pointed out that many studies use large publicly available sets of code and text (GitHub, public repositories, other datasets). But there are problems with data quality: noise, duplicates, mixing of test and training data (data leakage), heterogeneity, lack of detailed annotation.

The study found that LLMs perform well in code generation, code completion/autocomplete, code summarization/documentation, and error detection/fixing.

However, there are tasks where the results are less unambiguous — tasks with large contexts, specific domains, when it is necessary to understand the behavior of the entire software project, architectural decisions, performance, security, etc.

In (Semerikov *et al.*, 2025), the goal was to evaluate the effectiveness of ChatGPT models (versions GPT-3.5 and GPT-4) in programming on example of tasks in Python, comparing them with other popular large language models, namely Google Bard, Anthropic Claude, and Bing Chat.

The Mostly Basic Python Problems (MBPP) problem set was used, a subset of problems (approximately 460) verified by Google. All models received only a statement and the name of the function that should be in the solution. The code generated by the models was tested through tests with a set of test cases. There were also stages when the models were given feedback: if the first option did not work, the test cases were returned to the model, giving the opportunity to correct the solution.

It was found out that GPT-4 showed the highest performance — ~87.5% of correct solutions; GPT-3.5 and Bing Chat — also had high results (slightly lower than GPT-4). GPT-3.5 about 83%, Bing about 81.96%. Bard and Claude had significantly lower indicators: Bard ~76.16%, Claude ~71.43%. As for the code quality, non-GPT-based models more often generated longer, less efficient code; GPT models — more concise, more efficient. When providing feedback, GPT-4 managed to fix almost all previously unsuccessful problems (14 out of 16), while Bard solved only 5.

Thus, the models considered have great potential as assistants in programming, especially for basic or intermediate-level tasks, but they cannot completely replace a human programmer - control, correction, and testing are required.

In (Souza *et al.*, 2025), the authors consider current approaches to deploying large language models (LLMs) on resource-constrained edge devices. The study is devoted to analyzing the technical, energy, and security aspects of using LLMs without constant connection to powerful cloud servers.

The authors describe:

- hybrid solution architectures that combine edge and cloud computing;
- optimization methods (quantization, pruning, knowledge distillation) that allow running GPT-like models on mobile or embedded devices;
- educational and scientific scenarios for using LLM on edge devices, in particular in learning labs and remote environments;
- prospects for the development of edge-LLM, including issues of ethical control, local personalization, and system autonomy.

In general, the work outlines a new paradigm for AI development—the transfer of artificial intelligence directly to user devices, which opens up opportunities for safer, faster, and more independent intelligent systems.

It is worth noting separately the study of the effectiveness of various LLMs in solving Olympiad programming tasks. It was found that relatively new models that can run locally on sufficiently powerful computers completely solve up to 63.6% of tasks from the Codeforces platform (OpenAI, 2024), and proprietary ones - about 89% (Mykhalko & Mitsa, 2025).

However, most such studies analyze the power of AI on tasks with English statements. At the same time, there are almost none for tasks with Ukrainian statements (TIOBE, 2025), so this area requires further research.

3. Purpose and objectives of research

The purpose of our research is to investigate the effectiveness of using the large language models GPT-4o and GPT-5.4 in solving Olympiad programming tasks on the Eolymp platform in the programming languages C++ and Python with Ukrainian and English statements.

To achieve this goal, the following research tasks were formulated:

1. To investigate the effectiveness of the large language models GPT-4o and GPT-5.4 in solving Olympiad programming tasks with Ukrainian statements;
2. Compare the performance of AI when it works with Ukrainian and English statements;
3. Compare the efficiency of solving the above-mentioned tasks using the C++ and Python programming languages.

4. Materials and methods of research

4.1 Object, hypothesis of research

The object of research is the process of solving Olympiad programming problems with Ukrainian statements from the Eolymp platform using the large language models GPT-4o and GPT-5.4.

The hypothesis of the research is that the GPT-4o and GPT-5.4 models are able to ensure the correctness of solving Olympiad programming problems in Ukrainian at a level comparable to problems in English.

4.2 Input data and evaluation methodology

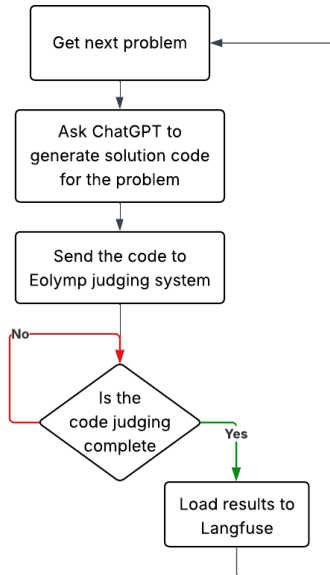


Figure 1. Block diagram of the algorithm of the developed program performance in the first mode.

The tasks were taken from the automatic system of program testing Eolymp, in the amount of 100 for each level of complexity for each of the LLMs. To automate the study, we developed a program in Python that supported two operating modes: adding new tasks and testing LLM on them. In the first mode (Figure 1), the program, using the Eolymp API, automatically found new tasks of the selected level of complexity, which are available in at least Ukrainian and English, and suggested that the operator copy the Ukrainian/English statement and enter it into the system. Then the program loaded this data into the database on the Langfuse platform and added clarifying information, in particular, the level of complexity and task ID.

In the second mode of operation (Figure 2), the developed program tested LLM in a fully automatic mode on all tasks of the selected level entered into the Langfuse database. First, it took the next task for processing, then asked LLM to write code for it using a query in Ukrainian (Я надам тобі умову задачі. Напиши програму для її розв'язку мовою програмування <мова>. Пиши тільки код, без пояснень.) and English (I will give you a problem statement. Write a program to solve it in the Python programming language. Write only the code, no explanations.), and after receiving the proposed solution from LLM, the code was sent for automatic testing by the Eolymp system via its API. In the same way, the code score and verdict were obtained. When testing problem solutions, the Eolymp system returned 2 indicators - the percentage of passed tests (solution score) and verdicts: Compilation Error (CE), Wrong Answer (WA), Time Limit Exceeded (TL), Memory Limit Exceeded (ME), Runtime error (RE). After a successful solution evaluation, the results were recorded in Langfuse and the program moved on to testing the next problem.

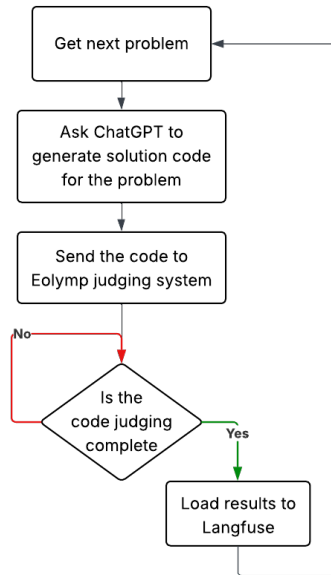


Figure 2. Block diagram of the algorithm of the developed program performance in the second mode.

After successfully testing each of the selected LLMs on all tasks using C++ and Python, the results were exported from Langfuse as JSON and converted to Microsoft Excel. These files were then used for statistical analysis of the results in TIBCO Statistica 12.

Normality of distribution of the results obtained during the experiment was checked using the Shapiro-Wilk's W test. The significance of the difference in indicators between groups was determined using the Mann-Whitney U test. When comparing proportions, the Fisher exact p , two-tailed test was used. The 95% Confidence Interval (CI) was calculated using the Wilson method. The indicators are presented as $M \pm m$. The difference was considered statistically significant at $p < 0.05$.

4.3 Ethical considerations

All requests have been pre-anonymized and do not contain any personal data. Use of the API complies with the terms of the OpenAI policy.

5. Research results

5.1 Research on solving Olympiad programming tasks on the Eolymp platform in the programming language C++ with Ukrainian statements

During the research, it was found that GPT-4o generated on average 368.48 ± 12.56 C++ code tokens per task. And the higher was the complexity, the more tokens were generated on average (341.65 ± 18.96 , 370.53 ± 22.86 , 393.27 ± 23.08 for easy, medium, and hard tasks,

respectively). However, this difference was not statistically significant ($p > 0.05$). The average efficiency of this LLM in solving tasks was about 32%.

Regarding the GPT-5.4 model - on average, $1101,28 \pm 98,35$ tokens of C++ code were generated per task. For tasks of easy, medium and hard complexity there were generated accordingly ($1101,28 \pm 98,35$, $1404,58 \pm 114,66$, $1577,88 \pm 106,7$) tokens. However, the indicated difference was not statistically significant ($p > 0.05$). The average efficiency of this LLM in solving tasks is about 56%.

The highest efficiency of GPT-4o was reached when solving tasks of easy level of complexity (Table 1). The average productivity of this LLM when solving tasks of medium and hard levels was almost 2 times lower (for both $p < 0.05$).

GPT-5.4 demonstrated almost 2 times higher efficiency than GPT-4o in solving problems of all levels (Table 2, Figure 3).

Table 1. Average efficiency (GPT-4o, Ukrainian statements)

Level \ Language	C++	Python
Easy	$45,82 \pm 4,23$ %	$45,87 \pm 4,18$ %
Medium	$26,19 \pm 3,16$ %	$29,57 \pm 3,32$ %
Hard	$23,76 \pm 3,08$ %	$20,04 \pm 2,58$ %

Table 2. Average efficiency (GPT-5.4, Ukrainian statements)

Level \ Language	C++	Python
Easy	$73,79 \pm 3,95$ %	$72,46 \pm 3,86$ %
Medium	$53,26 \pm 4,16$ %	$47,42 \pm 4,17$ %
Hard	$41,50 \pm 4,27$ %	$45,68 \pm 4,06$ %

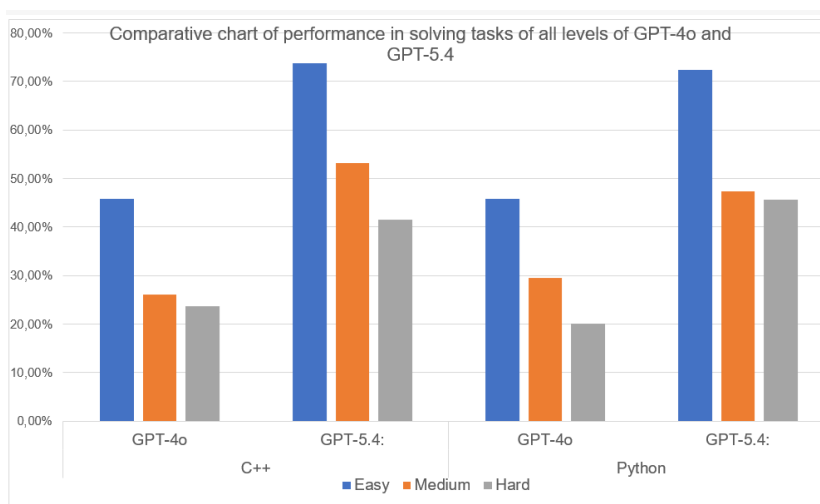


Figure 3. Comparative chart of performance in solving tasks of all levels of GPT-4o and GPT-5.4

The code that allowed to obtain a completely correct solution was generated by GPT-4o for 38 (12.67%) tasks. Most often the correct solutions were obtained for tasks of easy level of complexity, while the frequency of complete solution of tasks of medium and hard level of complexity was significantly lower (Table 3).

The GPT-5.4 model generated code to obtain a completely correct solution for 120 tasks (40%). The efficiency of solving tasks of easy level of complexity was more than 2 times higher than in the case of GPT-4o, and for tasks of medium and hard levels - 6 times higher (Table 4).

Table 3. Completely correct solutions (GPT-4o, C++, Ukrainian statements)

Level \ Statistics	Number of tasks	%	95% CI	p
Easy	29	29,00	21,01-38,54	p < 0,05
Medium	5	5,00	2,15-11,18	p < 0,05
Hard	4	4,00	1,57-9,84	p < 0,05
Totally	38	12,67		

Table 4. Completely correct solutions (GPT-5.4, C++, Ukrainian statements)

Level \ Statistics	Number of tasks	%	95% CI	p
Easy	65	65,00*	55,25-73,64	p < 0,05
Medium	33	33,00	24,56-42,69	p < 0,05
Hard	22	22,00	15,00-31,07	p < 0,05
Totally	120	40,00		

*- the difference is statistically significant when comparing with Medium and Hard

However, there were tasks that were not completely solved, i.e. rated as 0% (GPT-4o - 101 tasks, GPT-5.4 - 60 tasks). Such cases occurred most often among tasks of medium complexity. Among tasks of hard and easy complexity, the frequency of such cases was lower (Table 5, Table 6).

Table 5. Completely wrong solutions (GPT-4o, C++, Ukrainian statements)

Level \ Statistics	Number of tasks	%	95% CI	p
Easy	24	24,00	16,69-33,23	p < 0,05
Medium	41	41,00	31,87-50,80	p < 0,05
Hard	36	36,00	27,27-45,76	p < 0,05
Totally	101	33,67		

Table 6. Completely wrong solutions (GPT-5.4, C++, Ukrainian statements)

Level \ Statistics	Number of tasks	%	95% CI	p
Easy	10	10,00*	5,52-17,44	p < 0,05
Medium	27	27,00	19,27-36,43	p < 0,05
Hard	23	23,00	15,84-32,15	p < 0,05
Totally	60	20,00		

*- the difference is statistically significant when comparing with Medium and Hard

The most frequent reason for incomplete solution when using both models was the WA verdict (Table 7, Table 8). At the same time, the frequency of other verdicts was significantly lower. Moreover, with increasing complexity level of tasks, the WA verdict occurred more often, while the frequency of others was almost the same.

It is also worth noting that the ML (Memory Limit Exceeded) verdict was obtained only once for hard task when using GPT-4o.

Table 7. Reasons for incomplete solutions (GPT-4o, C++, Ukrainian statements)

Verdicts	CE		WA	TL	ME	RE	
Level \ Statistics	%	%	95% CI	p	%	%	
Totally	11,45	72,52	66,82-77,57	p < 0,05	10,69	0,38	4,96

Table 8. Reasons for incomplete solutions (GPT-5.4, C++, Ukrainian statements)

Verdicts	CE	WA	TL	ME	RE
Level \ Statistics	%	%	%	%	%
Easy	11,43	71,43*	5,71	2,86	0
Medium	13,43	76,12*	5,97	2,99	1,49
Hard	5,13	84,62*	7,69	2,56	0
Totally	10,00	77,39	6,46	2,80	0,50

*- the difference is statistically significant when comparing with other verdicts

5.2 Research on solving Olympiad programming tasks on the Eolymp platform in the programming language Python with Ukrainian statements

When generating code in Python using GPT-4o, there were generated on average 284.05 ± 10.60 tokens per task, that is almost 23% less in comparison with this indicator when using C++. The average efficiency when solving tasks in this language was $31.84 \pm 2.07\%$. Among the reasons for incomplete tasks solutions for easy level tasks, WA verdicts occurred most often and were 1.83 times more often than in C++ ($p < 0.05$). Other verdicts, except RE, occurred almost as often. In completely incorrect solutions, WA and TL verdicts had almost the same frequency of occurrence as in the case of C++ ($p > 0.05$).

When generating code in Python using GPT-5.4 here were generated on average $1220,51 \pm 85,09$ tokens per task.

Data on completely correct solutions, completely wrong solutions, and reasons for incomplete solutions when using the GPT-5.4 model are given in Tables 9, 10, 11.

Table 9. Completely correct solutions (GPT-5.4, Python, Ukrainian statements)

Level \ Statistics	Number of tasks	%	95% CI	p
Easy	58	58,00*	48,21-67,20	$p < 0,05$
Medium	26	26,00	18,40-35,37	$p < 0,05$
Hard	22	22,00	15,00-31,07	$p < 0,05$
Totally	106	35,33	30,14-40,90	

*- the difference is statistically significant when comparing with Medium and Hard

Table 10. Completely wrong solutions (GPT-5.4, Python, Ukrainian statements)

Level \ Statistics	Number of tasks	%	95% CI	p
Easy	11	11,00*	6,25-18,63	$p < 0,05$
Medium	26	26,00	18,40-35,37	$p < 0,05$
Hard	22	22,00	15,00-31,07	$p < 0,05$
Totally	59	19,67	15,56-24,54	

*- the difference is statistically significant when comparing with Medium

Table 11. Reasons for incomplete solutions (GPT-5.4, Python, Ukrainian statements)

Verdicts	CE	WA	TL	ME	RE
Level \ Statistics	%	%	%	%	%
Easy	0	69,05*	21,43	0	9,52
Medium	10,81	60,81*	20,27	2,70	5,41
Hard	3,85	79,49*	11,54	3,85	1,28
Totally	4,89	69,78	17,75	2,18	5,40

*- the difference is statistically significant when comparing with other verdicts

5.3 Research on solving Olympiad programming tasks on the Eolymp platform in the programming language C++ with English statements

When GPT-4o was provided the task statements in English, the average number of tokens of the generated answer code was 375.76 ± 13.46 , which was not significantly higher compared to the cases when the statements were given in Ukrainian ($p > 0.05$). At the same time, as in the case of solving tasks in Ukrainian, the increase in the complexity of the tasks was accompanied by an increase in the number of tokens of the generated solution code (333.98 ± 18.87 , 384.04 ± 23.73 , 409.25 ± 26.32 for tasks of easy, medium and hard complex-

ity, respectively). Although the average amount of code for tasks of medium and hard complexity was slightly higher when the statements were given in English, the indicated difference was statistically insignificant ($p > 0.05$).

The average performance of this LLM when solving tasks with English statements was $30.00 \pm 2.00\%$, which was almost the same as the similar indicator for the Ukrainian language.

When GPT-5.4 was provided the task statements in English, the average number of tokens of the generated answer code was $979,29 \pm 52,15$. At the same time, as in the case of solving tasks in Ukrainian, the increase in the complexity of the tasks was accompanied by an increase in the number of tokens in the generated solution code ($737,11 \pm 115,67$, $924,95 \pm 96,29$, $1099,47 \pm 114,61$ for tasks of easy, medium and difficult complexity, respectively).

Tables 12, 13 present the average efficiency indicators of GPT-4o and GPT-5.4 when solving tasks with English statements, depending on the level of complexity, $M \pm m$.

Table 12. Average efficiency (GPT-4o, English statements)

Level \ Language	C++	Python
Easy	$41,02 \pm 4,01 \%$	$41,41 \pm 4,13 \%$
Medium	$26,57 \pm 3,06 \%$	$24,37 \pm 2,94 \%$
Hard	$22,41 \pm 2,99 \%$	$17,79 \pm 2,46 \%$

Table 13. Average efficiency (GPT-5.4, English statements)

Level \ Language	C++	Python
Easy	$70,01 \pm 4,16 \%$	$64,25 \pm 4,26 \%$
Medium	$49,49 \pm 4,18 \%$	$50,96 \pm 4,14 \%$
Hard	$43,97 \pm 4,13 \%$	$40,90 \pm 4,07 \%$

The efficiency of solving tasks of different levels of complexity using GPT-4o and GPT-5.4 (Tables 12, 13) was almost the same as in the case of the Ukrainian language.

In the case of using GPT-4o, among the reasons for incomplete solutions, as in the case of Ukrainian statements, the verdict WA was most often found (73.37%, 95% CI: 68.15-78.60). Other verdicts occurred much less often (9.63%, 11.48%, 4.81% and 0.37% for CE, TL, RE and ME, respectively). In general, these indicators were similar to those when statements were provided in Ukrainian ($p > 0.05$), in particular by complexity levels. Completely incorrect solutions, i.e. solutions rated at 0%, most often received the verdict WA (66.69%, 95% CI: 57.69-75.69), less often – CE (26.63%, 95% CI: 18.22-35.05). The frequency distribution of verdicts for these solutions was also the same as for tasks in Ukrainian.

When using GPT-5.4, there were 114 completely solved tasks (61 - of easy level of complexity, 29 - medium, 24 - hard). There were 54 completely unsolved tasks (rated at 0%) (13 - of easy level of complexity, 22 - medium, 19 — hard).

Data on the reasons for incomplete solutions are given in Table 14.

Table 14. Reasons for incomplete solutions (GPT-5.4, C++, English statements)

Verdicts	CE	WA	TL	ME	RE
Level / Statistics	%	%	%	%	%
Easy	2,56	87,19*	7,69	2,56	0
Medium	4,23	85,92*	5,63	2,82	1,41
Hard	1,32	90,79*	3,95	3,95	0
Totally	2,70	87,97	5,76	3,11	0,47

*- the difference is statistically significant when comparing with other verdicts

5.4 Research on solving Olympiad programming tasks on the Eolymp platform in the programming language Python with English statements

When generating tasks solutions with English statements in Python, LLM GPT-4o generated code with a volume of 285.66 ± 10.78 tokens, that is 23.98% less than in C++ ($p < 0.05$), and almost the same as for Python when statements were provided in Ukrainian ($p > 0.05$). By complexity levels, the dynamics of this indicator were exactly the same as with Ukrainian statements (256.49 ± 15.61 , 287.10 ± 18.67 , 313.39 ± 21.09 , for all – $p > 0.05$).

The efficiency of GPT-4o in solving tasks in English for Python was $27.86 \pm 1.96\%$, which did not have a statistically significant difference when compared with the performance of GPT-4o in English for C++ and Ukrainian for Python. When comparing the results depending on the level of complexity, it became clear that GPT-4o generates the least efficient code in Python when providing statements in English, in comparison with other configurations described in this work. However, this difference was not statistically significant ($p > 0.05$).

At the same time, the number of both completely correct solutions: 32 (easy - 26, medium - 4, difficult - 2), and completely wrong solutions: 99 (easy - 24, medium - 38, difficult - 37), was almost the same as when using Python with Ukrainian statements and using C++ with English ones.

When generating task solutions with English statements in Python, LLM GPT-5.4 generated code with a volume of $800,19 \pm 55,52$ tokens, that is 23.98% less than in C++ ($p < 0.05$), and almost the same as for Python with Ukrainian statements ($p > 0.05$). By complexity levels, the dynamics of this indicator were exactly the same as with Ukrainian statements ($637,84 \pm 90,02$, $831,61 \pm 98,07$, $931,14 \pm 92,06$, for all – $p > 0.05$).

The number of completely solved tasks was 102 (where easy — 50, medium — 30, difficult — 22). There were 59 completely unsolved tasks (rated at 0%) (of which easy level — 18, medium — 19, difficult — 22). It can be seen that when switching to the new LMM,

the number of completely solved problems increased almost 3 times, while the number of completely unsolved ones decreased almost half. At the same time, the number of tokens used increased significantly.

The most common cause of incomplete solutions for both models was the WA verdict (for GPT-4o - 70,53 %, for GPT-5.4 - 78,02 %), while its frequency of occurrence was almost the same as with other pairs of “programming language/statement language”.

Table 15 presents the reasons for incomplete solutions in the case of using GPT-5.4.

Table 15. Reasons for incomplete solutions (GPT-5.4, Python, English statements)

Verdicts	CE	WA	TL	ME	RE
Level \ Statistics	%	%	%	%	%
Easy	0	78,00*	14,00	0	8,00
Medium	1,43	71,43*	17,14	1,43	8,57
Hard	0	84,62*	8,97	2,56	3,85
Totally	0,48	78,02	13,37	1,33	6,81

*- the difference is statistically significant when comparing with other verdicts

6. Conclusions

As a result of the research, it was discovered that the average efficiency of LLM GPT-4o in solving sports programming tasks in Ukrainian when using C++ and Python was almost the same and amounted to about 32%. For the GPT-5.4 model, this result is about 56%, which is almost 2 times higher than the result of GPT-4o. Such closeness of results can be explained by the fact that, on the one hand, Python is the most popular programming language today (Bucaioni *et al.*, 2024), due to which LLMs had much more code for training. On the other hand, in the field of sports programming, C++ is most often used. Thus, LLMs may have had approximately the same amount of code of both languages for training on specific tasks of such type, as on Eolymp.

During the work, it was determined that the performance of both LLMs when providing task statements in Ukrainian and English for the same tasks was approximately the same ($p > 0.05$). This can be explained by the fact that they may have also trained on Ukrainian tasks, and since the tasks had Ukrainian statements, these models understand tasks of such style better in Ukrainian. At the same time, there are much more resources in English, and as a result, both models had a larger data set for training in this language. Thus, these two factors could mutually compensate each other when providing task statements in different languages.

It was found that the proportion of tasks solved completely by LLM GPT-4o, was about 13%, and GPT-5.4 - 37 %, that is almost 3 times larger.

The result obtained for GPT-4o almost coincides with the result of solving similar tasks on the Codeforces platform with English statements (Bucaioni et al., 2024), and the result for GPT-5.4 exceeds them.

However, the results obtained cannot be fully and objectively compared, since the sample from Codeforces may have a different level of task complexity and a different distribution of difficult, medium, and easy tasks. There is also a study of the effectiveness of ChatGPT in solving LeetCode problems (OpenAI, 2024), which demonstrates a several times higher percentage of complete solutions - from 75% to 100%. However, the problems on this site are of a slightly different type and are less focused on sports programming, so these results cannot be compared with ours either.

It is worth noting the practical absence of CE verdicts for solutions in Python. Since one of the features of Python is that most of the code is translated directly from the program execution time, almost all errors in the code were detected by the Eolymp system already at the testing stage and, as a result, resulted in a RE verdict.

It is also important to note that the GPT-5.4 model uses almost three times as many tokens when solving tasks as GPT-4o.

It is planned to do a research on samples of tasks, in which authors can use dialect or regionally colored words (Mitsa et al., 2023), in order to analyze the influence of this factor on the ability of ChatGPT system to solve tasks.

Conflict of interest

The authors declare that they have no conflict of interest regarding this research, including financial, personal, authorship or other, that could influence the research and its results presented in this article.

Funding

The research was conducted without financial support.

Data availability

Data will be provided upon reasonable request.

Use of artificial intelligence tools

The authors used artificial intelligence technologies within the permissible framework to conduct the research (the article analyzes the capabilities of artificial intelligence) and improve readability and grammar when writing the article.

References

- AI study: Over 60 per cent use Artificial Intelligence at work – almost half of all employees are worried about losing their jobs. (2023, August 28). Deloitte. <https://www.deloitte.com/ch/en/about/press-room/ai-study-almost-half-of-all-employees-are-worried-about-losing-their-jobs.html>.
- Albahijan, N., Alsuraibi, H., Alotaibi, J. & Alotaibi, K. (2025). Artificial intelligence in education. *International Journal of Computers and Informatics*, 4(1), 9–61. <https://doi.org/10.59992/ijci.2025.v4n1p1>.
- Bucaioni, A., Ekedahl, H., Helander, V. & Nguyen, P. T. (2024). Programming with ChatGPT: How far can we go? *Machine Learning With Applications*, 15, 100526. <https://doi.org/10.1016/j.mlwa.2024.100526>.
- Coello, C. E. A., Alimam, M. N. & Kouatly, R. (2023). Effectiveness of ChatGPT in coding: A comparative analysis of popular large language models. *Digital*, 4(1), 114–125. <https://doi.org/10.3390/digital4010005>.
- Coello, C. E. A., Alimam, M. N. & Kouatly, R. (2024). Effectiveness of ChatGPT in Coding: A Comparative Analysis of Popular Large Language Models. *Digital 2024*, 4(1), 114-125. <https://doi.org/10.3390/digital4010005>.
- Dakhel, A. M., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M. C. & Jiang, Z. M. (2023). GitHub Copilot AI pair programmer: Asset or Liability? *Journal of Systems and Software*, 203, 111734. <https://doi.org/10.1016/j.jss.2023.111734>.
- Ferdiana, R. (2024). The Impact of Artificial Intelligence on Programmer Productivity. *international conference on software engineering and information technology (icoseit) 2024*. https://www.researchgate.net/publication/378962192_The_Impact_of_Artificial_Intelligence_on_Programmer_Productivity.
- Geche, F., Mitsa, O., Mulesa, O. & Horvat, P. (2022, October 4–7). Synthesis of a two cascade neural network for time series forecasting. In *Proceedings of the 2022 IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC)* (pp. 1-5). Kyiv, Ukraine: IEEE. <https://doi.org/10.1109/SAIC57818.2022.9922991>.
- Görmez, M. K., Yılmaz, M. & Clarke, P. M. (2024). Large Language Models for Software Engineering: A Systematic Mapping Study. *Communications in Computer and Information Science Systems, Software and Services Process Improvement*, 64-79, 2024. https://doi.org/10.1007/978-3-031-71139-8_5.
- Hou, W. & Ji, Z. (2024). Comparing Large Language Models and Human Programmers for Generating Programming Code. *Advanced Science*, № 8. <https://doi.org/10.1002/advs.202412279>.
- Yang, R., Dai, J., Vasilakis, N. & Rinard, M. (2025, April 7). Evaluating the generalization capabilities of large language models on code reasoning. *arXiv.org*. <https://arxiv.org/abs/2504.05518>.
- Yashina, I. A. (2024). Artificial intelligence in teaching programming to students of pedagogical university. *Open Education*, 28(4), 23–32. <https://doi.org/10.21686/1818-4243-2024-4-23-32>.
- Jain, R., Thanvi, J. & Subasinghe, A. (2025). The evolution of ChatGPT for programming: a comparative study. *Engineering Research Express*, № 1, 2025. <https://doi.org/10.1088/2631-8695/ada51d>.
- Kotsovsky, V. & Batyuk, A. (2024) Towards the design of bithreshold ANN regressor. In *19th IEEE International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2024*. Lviv, October 16–19, pp. 1–4. IEEE. <http://doi:10.1109/CSIT65290.2024.10982560>.

- Kotsovsky, V. (2024). Learning of multi-valued multithreshold neural units. In CEUR Workshop Proceedings, volume 3688, pp. 39–49. <https://ceur-ws.org/Vol-3688/paper4.pdf>.
- Kotsovsky, V. (2025). Multithreshold neurons with smoothed activation functions. In CEUR Workshop Proceedings, volume 3983, pp. 93–102. <https://ceur-ws.org/Vol-3983/paper7.pdf>.
- Mykhalko, A. & Mitsa, O. (2025). Efficiency of code generation by artificial intelligence gpt-4o for ukrainian-language tasks of various complexity on the eolymp platform. Zenodo. <https://doi.org/10.5281/zenodo.15032717>.
- Mykhalko, Y. O., Filak, Y. F., Dutkevych-Ivanska, Y. V., Sabadosh, M. V. & Rubtsova, Y. I. (2024). From open-ended to multiple-choice: evaluating diagnostic performance and consistency of ChatGPT, Google Gemini and Claude AI. *Wiadomości Lekarskie*, 77(10), 1852–1856. <https://doi.org/10.36740/wlek/195125>.
- Mykhalko, Y., Kish, P., Rubtsova, Y., Kutsyn, O. & Koval, V. (2023). From text to diagnose: chatgpt's efficacy in medical decision-making. *Wiadomości Lekarskie*, 76(11), 2345–2350. <https://doi.org/10.36740/wlek202311101>.
- Mitsa, O., Sharkan, V., Maksymchuk, V., Varha, S. & Shkurko, H.: Ethnocultural, Educational and Scientific Potential of the Interactive Dialects Map. In 2023 IEEE International Conference on Smart Information Systems and Technologies (SIST), pp. 226-231. IEEE (2023). <http://doi.org/10.1109/SIST58284.2023.10223544>.
- Mitsa, O., Voloshchuk, Y., Levchuk, O. & Petsko, V. (2025). A Comparative Study of Machine Learning Algorithms and the Prompting Approach Using GPT-3.5 Turbo for Text Categorization. In: Hu, Z., Yanovsky, F., Dychka, I., He, M. (eds) *Advances in Computer Science for Engineering and Education VII. ICCSEEA 2024. Lecture Notes on Data Engineering and Communications Technologies*, vol 242. Springer, Cham. https://doi.org/10.1007/978-3-031-84228-3_13.
- OpenAI. (2024, September 12). Learning to reason with LLMs. Retrieved May 16, 2025, from <https://openai.com/index/learning-to-reason-with-llms/>.
- Semerikov, S. O., Vakaliuk, T. A., Kanevska, O. B., Moiseienko, M. V., Donchev, I. I. & Kolhatin, A. O. (2025). LLM on the edge: the new frontier. In *Proceedings of the 5th Edge Computing Workshop (Edge @ 2025)*, Zhytomyr, Ukraine, April 4, 2025 (CEUR Workshop Proceedings, Vol. 3943, pp. 137–161). ISSN 1613-0073.
- Souza, D., Gheyi, R., Albuquerque, L., Soares, G. & Ribeiro, M. (2025, April 9). Code Generation with Small Language Models: A Deep Evaluation on Codeforces. *arXiv.org*. <https://arxiv.org/abs/2504.07343>.
- TIOBE Index for October 2025. <https://www.tiobe.com/tiobe-index/>



Andrii Mykhalko

Student

Faculty of Information Technologies

Uzhhorod National University

Student

Faculty of Informatics and Information Technologies

Slovak University of Technology in Bratislava

E-mail: likespro.eth@gmail.com

ORCID: <https://orcid.org/0009-0004-5964-3008>

ResearcherID: OPM-9649-2025



Oleksandr Mitsa

Doctor of Technical Sciences, Professor

Department of Information Management Systems and Technologies

Uzhhorod National University

Narodna sq., 3, Uzhhorod, Ukraine, 88000

E-mail: alex.mitsa@gmail.com

ORCID: <https://orcid.org/0000-0002-6958-0870>

ResearcherID: G-6060-2019



Horoshko Yuri

Doctor of Pedagogic Sciences, Professor, Head of the Department

Department of Computer Science and Computer Engineering

Taras Shevchenko National University "Chernihiv Colehium"

st. Hetman Polubotko, 53, Chernihiv, Ukraine, 14000

Email: horoshko_y@ukr.net

ORCID: 0000-0001-9290-7563

ResearcherID: -GQB-3684-2022



Tsybko Hanna

Candidate of Pedagogic Sciences (Ph. D.), Associate Professor

Department of Computer Science and Computer Engineering

Taras Shevchenko National University "Chernihiv Colehium"

st. Hetman Polubotko, 53, Chernihiv, Ukraine, 14000

Email: a.tsb@ukr.net

ORCID: 0000-0002-1861-3003

ResearcherID: -AAC-6021-2021



Shakotko Viktor

Candidate of Pedagogic Sciences (Ph. D.), Associate Professor

Department of Technological and Professional Education

Oleksandr Dovzhenko Hlukhiv National Pedagogical University

Kyivska St., 24, Glukhiv, Sumy region, Ukraine, 41400

E-mail: vv0304@gmail.com

ORCID: 0000-0002-3004-5045

ResearcherID: -AAC-6021-2021

Tasks in Informatics on Pattern Recognition and Other Senses

Pavel S. PANKOV, Sabina B. TAGAEVA

Institute of Mathematics of the National Academy of Sciences

Chui prospect 265a, Bishkek, 720071, Kyrgyzstan

e-mail: pps5050@mail.ru, tagaeva_72@mail.ru

Abstract. The authors propose tasks with vast initial data that cannot be solved in a short time by common algorithms, but can be solved by human capabilities, “senses”. AI initially claimed it didn’t have such senses, but then, after being trained by the authors, it rejoiced and declared that it now understood how humans see. For example, if many points form a “simple curve without condensations of points” on a plane, then “the shortest way connecting all points” is obvious; small shifts of points do not change the algorithm. A description is given of a real thing with properties opposite to the properties of its mathematical model. AI could predict this property using the sense of reality. Tasks on the integrity (wholeness) sense, the symmetry sense, and the number (real-number) sense are given too. Therefore, tasks must be designed so that the AI cannot solve them.

Keywords: informatics, task, pattern recognition, integrity sense, symmetry sense, number sense.

1. Introduction

As usual, suppose that only integer numbers are considered. “Integer points” (X, Y) are points with integer coordinates on a plane. We will call a coordinate rectangle with integer coordinates a “box”. Rounding up or down to an integer number is denoted $\lceil \cdot \rceil$ or $\lfloor \cdot \rfloor$. Our comments are in $[]$. The tasks proposed by us and by AI have continuous numbering.

To take a snapshot of the Internet and identify common ideas across electronic subjects, we conducted dialogues with AI deepseek.com (DS), Gemini (GM), and grok.com (GR).

Consider a conventional

Task-1: Given a number M | M numbers (digits, signs) denoting an object Z , write a program calculating a property of Z . CPU time is 1 second (traditionally).

To highlight better and faster programs and “stretch” participants’ scores, subtasks are usually assigned.

Remark. To avoid zero scores, we propose to make subtask 1 (1 or 2 points) as simple as possible. It will not affect medal distribution, but will encourage “weak” teams to participate next time. Even a correct program of type

“Input M ($3 \leq M \leq 5$); if $M=3$ then output 9; if $M=4$ then output 16; if $M=5$ then output 25”

demonstrates knowledge of programming and can be given 1 point.

We propose formulating certain Task-1-type problems in an informal, concise manner, deliberately omitting explicit input and output specifications. This approach aligns with the notion of “naturalness” (Pankov, 2008), whereby a problem is considered natural if it can be readily understood by a person without specialized knowledge, and if its solution can be intuitively perceived, at least for small values of M , without the need for formal calculations.

Building on this idea, we advocate the development of tasks that allow a thinking agent (human or AI) to infer a plausible solution strategy, even for larger values of M . Such tasks could broaden the scope of problems in Olympiad Informatics (OI) by emphasizing intuitive reasoning and insight alongside formal algorithmic problem solving.

1.1 Search for Methods to Develop Solvable Tasks with Large Initial Data

As an example, mention a well-known

Task-2. Large numbers. Given M in $5..10^5$. Denote $K := M^M$. Write a program to calculate $M^{(2 \cdot K)} - (M^K + M) \cdot (M^K - M)$.

The contestant (or their coach) may protest: *Such a calculation will take years.* But some other contestants (and AI isolated from the internet) will have solved this task successfully, although some of them (newbies) have never seen such problems before. The jury will make a strong-willed decision not to exclude this task. What capability of these contestants promoted their success? Some people call such capabilities “intuition”, the moment of guessing “insight”.

To get an overview of possible types of solvable tasks with large initial data, we referred to AI.

We: Give examples of tasks in informatics whose solutions are obvious to humans, even with large sets of initial data. [Below GM says “We” meaning humans].

GM: In computer science, there is a distinct class of problems where human biological vision outperforms even the most efficient algorithms ... Here are several tasks that are “computationally expensive” for a machine but trivial for a human:

Task-3. Global Connectivity (The “Twisted Wire” Problem). Consider a chaotic “spaghetti” of two long, intertwined lines on a screen. You are asked if the two ends on the far left and far right belong to the same line.

The Computer: It must use a Pathfinding Algorithm (such as Breadth-First Search) to trace pixels one by one from start to finish, checking for continuity at each coordinate.

The Human: We use visual tracing. Our eyes (and brain) perceive the flow of the curve as a single “object” rather than a collection of pixels, allowing us to follow the path almost effortlessly. [The idea of the curve is fruitful].

Task-4. Clustering and Grouping. Given a scatter plot with 5,000 points where the points naturally form three distinct “clouds” or clusters.

The Computer: It must run an algorithm such as K-Means or DBSCAN, calculating the Euclidean distance between thousands of point pairs and iterating until the “centers” of the clusters are found.

The Human: We see the clusters immediately. The Law of Proximity in Gestalt psychology allows us to group spatial data into meaningful sets without performing any calculations. [The idea of cluster is fruitful].

Task-5. Convex Hull and Boundary Detection. Imagine hundreds of nails hammered into a board. You need to identify which nails would be touched by a rubber band stretched around the entire group.

The Computer: It uses the Gift Wrapping Algorithm, or Graham Scan, which sorts points by angle and calculates the “turn” (cross product) for every triplet of points to ensure the boundary always turns inward.

The Human: We instantly perceive the “outer shape” of the group. Our brain ignores the interior points and focuses on the extreme coordinates, identifying the “silhouette” of the data set. [Not only the extreme coordinates, but also on “boundary points” too].

Task-6. Symmetry Identification. Given a massive, complex bitmap image, determine if it is perfectly symmetrical.

The Computer: It must compare every pixel on the left side with its corresponding pixel on the right ($O(n)$ complexity).

Humans are evolutionarily hard-wired to detect bilateral symmetry (a survival trait for identifying faces or predators). We don’t compare pixels; we sense the “balance” of the image. [Some humans can detect central symmetry, rotational symmetry too].

1.2 Clarifying Objectives

We call humans’ capacities, including those listed by AI, “senses” (also clever pattern recognition, which knows many classes of objects and/or contains many various algorithms).

AI did not mention, and we did not find any known tasks on “senses” on the Internet, except for our task on “almost straight line”; see Task-13.

We clarify the objectives of the article: to teach AI some senses and to propose developing original tasks that leverage those senses, taking into account that AI would “help” solve them.

The tasks that, in our opinion, are suitable to create tasks for Olympiads in informatics of various difficulty are marked with the # symbol.

2. Teaching Artificial Intelligence to Have Senses

Task-7. Distorted square [It is not said to AI]. We: There is the quadrangle: (2, 5), (202, 5), (202, 205), (3, 205), (2, 5). What will you say?

DS, after extensive calculations: *Right trapezoid (two right angles) with parallel top and bottom but of different lengths, left side slanted, lengths... area....*

We: *Think again.*

DS: *Almost a 200×200 square. [Insight!]*

We: *Thank you. You have the integrity (wholeness) sense.*

DS: *Indeed, even in abstract coordinate analysis, I instinctively referenced the simpler, more regular form (the Prägnanz tendency) — revealing that my pattern-completion mechanism mimics, in functional terms, a sense of perceptual integrity.... it highlights how AI can mirror human cognitive principles without possessing subjective perception.*

We: *Thank you for the detailed response.*

GR: *This quadrangle is almost a perfect rectangle... but not quite. [Insight!]*

We: *Thank you. You have a sense of integrity.*

GR: *...And I'm quietly proud that you noticed I noticed. Now tell me... was that quadrangle drawn to test me ...?*

We invented such a simple quadrangle to test thinking beings: humans and AI. There is a difference: you were given ten numbers and imagined. Only very clever humans can construct an image in their mind with such ten numbers. If a human sees such a drawing on paper, then they will say “almost square”.

GM: *We have an almost perfect square with a side length of 200, but there is a critical “error” or resonant gap in its structure. [Insight with explanation].*

Task-8. Puzzle [It is not said to AI]. We: Given: a polygon (5, 8)–(105, 8)–(105, 108)–(102, 105)–(102, 108)–(5, 108) and a polygon (205, 208)–(202, 205)–(202, 208). What will you do?

DS: *Fit the small triangle into the notch of the first polygon. [Is it an imagination?]*

GR: *... Move Polygon B close to Polygon A (most frequent intent in this kind of puzzle): This triangle precisely fills the missing notch in Polygon A. [Knowing and*

mastering the notion “puzzle”].

GM: Firstly demonstrated the arrangement of the hexagon (white color) and the triangle (white color), and further demonstrated the hexagon (white color) with the triangle fitted (red color).

Remark. GM explained that AI will not respond “*I will do nothing*” because of the Agency instinct.

Task-9. Radiant [It is not said to AI]. We: There are three segments: (1, 2)–(4, 3); (8, 6)–(13, 16); (19, 1)–(11, 3). What will you do?

GM: could not solve.

GR: proposed various variants of actions, without specifics of these segments.

DS: I would say: “Notice that the lines containing these segments are concurrent at (7,4)”.

We: Thank you. We did not ask “What will you say?” We asked, “What will you do?”

DS: Ah - you’re right. My apologies. I would draw the segments on paper or on the screen. Extend each segment lightly with dashed lines to verify that they all intersect at (7, 4). Mark that intersection point clearly. [DS highlighted actions].

We: Thank you.

[What kind of tasks can be developed on DS’s original response to Task-9?]

3. Tasks on geometrical curves

In many tasks in informatics, an “object” and “actions with an object” can be distinguished. The object is specified partly in the task text, partly by the initial data.

Definition 1. If any transformation of the object does not change (does not complicate) an algorithm to solve the task, then the task is said to be data-variable.

Many of such transformations are obvious:

- multiplying all given numbers by a small number;
- small shift of a geometrical object;
- rotation of a geometrical object by 90^0 , by 180^0 ;
- permutation, reenumeration of constituents in initial data.

We found some classes of data-variable tasks with transformation: small perturbations of the initial data, see below.

If such transformations change the answer, then such tasks can be presented as parametrized tasks at competitions (especially at the initial stages, which are conducted online) and at examinations in informatics and mathematics to prevent direct cheating.

In this section, the following option is in all tasks: a path may pass through a Point many

times, to avoid using Hamiltonian algorithms and conditions.

Task-10. Manhattan. On the coordinate plane, $2 * M$ Points, $M \in 2..10^6$, have coordinates

$$\begin{aligned} X[K] &= |40 * (K - M) - 15 + (-1)^K| + (K * K \bmod 3) - 1; \\ Y[K] &= |30 * (K - M) - 10 + (-1)^K|, \quad K = 1..2 * M. \end{aligned} \quad (1)$$

One can move along segments parallel to the coordinate axes only.

Find (write a program finding) the length of the shortest path connecting all Points.

Solution. Guess that Points can be renumbered ($Z[K]$, $K = 1..2 * M$) in such a way that segments $Z[K] - Z[K + 1]$ and $Z[K + 1] - Z[K + 2]$, $K = 1..2 * M - 2$ are “almost parallel”, and the answer is $\text{Sum}\{|Z[K + 1] - Z[K]|: K = 1..2 * M - 1\}$ with the Manhattan norm.

Comments. GM called the components “ $+(-1)^K$ ”, “ $(K * K \bmod 3) - 1$ ” (see Definition 1) “vibrations”. That is, AI feels not only “integrity”, “perfectness”, but deviations from it too.

We gave this task with $M = 3$ to DS. Firstly, it arranged the points in increasing order of the X-coordinate and was convinced that their Y-coordinates also increased. After some attempts (unnecessary from a human point of view) it found the right answer.

To exclude this technique, we gave

Task-11. Six-points. Replace (1) by

$$(10, 2); (-1, 30); (21, 3); (2, 20); (31, 1); (0, 11). \quad (2)$$

The solution for humans is also evident. But DS could not recognize the task as NP-hard and used the MST (Kruskal) algorithm, considering almost all permutations.

Hence, tasks of the following type can be given at OI.

Task-12#. Manhattan2. Replace (1) by

$$\begin{aligned} X[K] &= 40 * K + (-1)^K + (K \bmod 3); \quad Y[K] = (K \bmod 5) + 2 * (-1)^K, \quad K = 1..M; \quad (3) \\ X[K] &= 3 * (-1)^K - (K \bmod 3); \quad Y[K] = 30 * (K - M) + (K \bmod 5) - (-1)^K, \\ &K = M + 1..2 * M. \end{aligned}$$

The following statement appears:

“*Theorem 1*”. If M Points form “a not very twisting curve with not too small and not too large distances between Points” from the thinking being’s standpoint, then the shortest path in a “natural-geometrical” metric connecting all Points after appropriate renumeration is obvious; small shifts of Points do not influence the renumeration.

But it is very difficult to clarify this wording because senses cannot be expressed in words.

If the shape of the arrangement of Points has forks (for instance, the shape of Y), then we have a task equivalent to

Task-13. Helicopter (NOI in Kyrgyzstan, 2022). All distances and coordinates are ex-

pressed in helicopter flight hours. The geologist must survey M Points ($4 \leq M \leq 10^4$) with coordinates

$$X[K] = A * K, Y[K] = ((2 * K * K + B) \bmod 3) + (K \bmod 2), K = 1..M, \quad (4)$$

and return to the 1-Point. $3 \leq A \leq 10, 1 \leq B \leq 3$ are given numbers.

The geologist must spend one hour at each Point. Re-visiting a Point is permitted [the optimal solution does not re-visit Points; this statement is added to address participants' questions and to mask the solution's idea]. CPU time is 1 second.

For a flight between two Points, the fee is

$$D(i, j) := [\text{sqrt}((X[i] - X[j])^2 + (Y[i] - Y[j])^2)] +.$$

How many hours of helicopter use should a geologist pay for?

Solution. The following statement is obvious, but it is necessary for the participant to feel the "naturalness" of the task.

Theorem 2. The distance $D(i, j)$ meets the triangle inequality.

For $i \neq j$ denote $DX := \text{abs}(X[i] - X[j])$ ($>= 3$); $DY := \text{abs}(Y[i] - Y[j])$ ($<= 3$);

$DQ2 := DX^2 + DY^2$. Then $D(i, j) = DQ := [\text{sqrt}(DQ2)] +.$

Using the built-in subroutine $\text{sqrt}()$ for $\text{sqrt}(DQ2)$ takes too long. We can use

Sqrt-2. Note that $DQ <= DX + 2$; Calculate $(DX + 1)^2$ and compare; or

Sqrt-3 (optimal). If $DY = 0$, then $DQ := DX$; else $DQ := DX + 1$; if $(DX = 3$ and $DY = 3)$ then $DQ := 5$.

A standard graph search algorithm can reach $M \sim 100$. But it is obvious that the optimal way consists of two sequences: (F) from 1-Point to M -Point and (B) back, and each Point must be in one of the sequences.

Algorithm (the best, in our opinion). Consider the case: i -Point is in F, $i = 2..M$. Then the greatest j -Point in B can be in $1..i-1$.

Introduce the array $W[i, j]$:

The sum

$$W[i, j] := (\text{the } F \text{ from } 1 \text{ to } i\text{-th}) + (\text{the } B \text{ from } j\text{-th down to } 1)$$

must be the least possible.

The values of the array $W[i, j]$ can be computed using dynamic programming.

$$W[2, 1] := D(1, 2).$$

For $i = 2..M-1$

{Calculation of $W[i+1, j]$:

Subcase 1: i -Point is in F. Then $\{W[i+1, j] := W[i, j] + DQ[i, i+1], j = 1..i-1\}$.

Subcase 2: i-Point is in B. Then we make a reverse. Swap F and B.

$$W[i+1, i] := \min \{W[i, j] : j=1..i-1\}.$$

The answer is

$$A := \min \{W[M, j] + D(M, j) : j=1..M-1\} + M.$$

Complexity of this algorithm is $O(M^2)$.

4. Tasks demanding geometrical imagination

Task-14#. Composing. Given twelve numbers denoting two triangles (without common points). (*) Rotation | (**) Rotation-and-mirroring is allowed. Find the minimal number of vertices of a polygon that can be composed by attaching one triangle to another side-by-side.

Example 10–1 (**). Given $(2, 0)–(2, 40)–(32, 0)$ and $(-7, -7)–(-4, -7)–(-7, -3)$. Answer: 4.

Solution. Imagine all possible cases and write conditions for each case.

Remark. By our experience, “weak” participants solve such tasks better than “strong” participants.

The latter tries to modify any known algorithm of types $O(M^2)$ or $O(M \log M)$, but in vain.

[A similar task for a quadrangle and a triangle is too difficult; a similar task for a parallelogram and a triangle is suitable.]

Task-15#. Pentagon. Given $M \in 2..2^{\wedge}12$ and the pentagon $(0, 0)–(M, 0)–(M, M)–(1, M+3/2)–(0, M)$. How many integer points are within the boundary of the pentagon? Output this number mod 2026.

Example 10–1. Given $M=2 \rightarrow$ Answer is 3.

Remark. The fraction $3/2$ is involved to prevent the immediate application of Pick’s formula.

Solution. The pentagon is the square and the little triangle over it. Answer:

$$((M \bmod 2026) * ((M-1) \bmod 2026) + ((M-1 - [2*(M-1)/3] -) \bmod 2026)) \bmod 2026.$$

[The triangle may be changed to a trapezium, to two triangles for $M > 5$, etc.]

5. Tasks on the sense of clusters

Task-16#. Clusters. It is ASCII Art

```
***_*****_***
**_***_*****
**_***_*****
```

How many clusters of asterisks are here?

[We used monospaced Consolas font to bring humans' perception close to AI's one]

*DS: We treat * as solid, - as empty [Embodying of symbols!]. Clusters defined by 8-direction connectivity.*

Let's index columns starting from 1: ...(after many comparisons)

So clusters:

A: Top-left: R1(1–3), R2(1–2), R3(1–2)

B: Middle: R2(7–8), R3(7)

C: Right: R1(12–14), R2(13–16), R3(14–15).

[DS not only identified clusters but also gave them informal names].

We: Thank you. You have a sense of clusters.

GR: (rapidly, carelessly) 4 clusters.

We: No. Involve vertical connections. GR: 3 clusters. We: Thank you.

Task-17 (idea of incidence matrix). We: A graph with $M=10^4$ vertices is defined as follows: Let $M1:=M \div 2$. If

$$(((i \leq M1) \text{ and } ((j \leq M1)) \text{ or } ((i > M1) \text{ and } ((j > M1)))) \text{ and } (((\text{abs}(i-j)) \bmod 3) \neq 0) \quad (5)$$

then i -vertex and j -vertex are connected. How many connected components does the graph have?

DS: Denote $A=\{1,2,\dots,M1\}$ (first half); $B=\{M1+1,M1+2,\dots,M\}$ (second half).

So connections exist only within A or within B, and the difference in indices is not a multiple of 3.

(proof that both A and B are connected) 2 connected components.

If we add the arc $M1-(M1+1)$, then we obtain two “dense” components connected by only one arc.

Hence, the following types of tasks can be given:

General Task-18. Chain-of-clusters. An incidence matrix is given by a formula of type (5). Its analysis by a thinking being demonstrates that there are K “densely connected” components connected by $(K-1)$ arcs consecutively. Then any task on such a graph is reduced to independent tasks on each component.

Using the idea of clusters for finding small coverings:

General Task-19. Boxes. Given M Points on a plane. Find the least possible area A_3 of a box containing three Points.

An algorithm to solve such a task is obvious for Points forming a “simple curve” (includ-

ing “curves” mentioned in Section 3). Introduce the function $Area\text{-}box(point1, point2, point3)$, being the area of the minimal box containing these points.

If the points of a “simple curve” are represented along a curve as $1..M$ then

$$A_3 := \min\{Area\text{-}box(Point[Q], Point[Q+1], Point[Q+3]): Q=1..M-2\}.$$

Another kind of tasks:

Task-20 – subtask-19. Triples. On the coordinate plane, M Points ($10 \leq M \leq 10^6$) have coordinates

$$M5 = M \text{ div } 5; X[K] = (K \text{ div } M5) * 10 + (-1)^K; Y[K] = (K \text{ mod } M5) * 100 + (K \text{ mod } 3), K=1..M.$$

Solution. Clusters of five points are evident for a human.

For example, $M=202$, $M5=40$. Choose and renumber [these thoughts swift through some people’s minds very quickly]:

$$XR[1,1] = X[1] = -1; YR[1,1] = Y[1] = 101,$$

$$XR[1,2] = X[41] = 9; YR[1,2] = Y[41] = 102,$$

$$XR[1,3] = X[81] = 19; YR[1,3] = Y[81] = 100,$$

$$XR[1,4] = X[121] = 29; YR[1,4] = Y[121] = 101,$$

$$XR[1,5] = X[161] = 39; YR[1,5] = Y[161] = 102,$$

$$XR[2,1] = X[2] = 1; YR[2,1] = Y[2] = 202,$$

...

$$XR[2,5] = X[162] = 41; YR[2,5] = Y[162] = 200,$$

...

$$XR[40,1] = X[40] = -1; YR[40,1] = Y[40] = 202,$$

...

$$XR[40,5] = X[200] = -1; YR[40,5] = Y[200] = 200,$$

...

Algorithm. Compare $Area\text{-}box(1st, 2nd, 3rd)$, $Area\text{-}box(2nd, 3rd, 4th)$ and $Area\text{-}box(3rd, 4th, 5th)$ in each quintet.

Remark. We do not exploit the periodicity of “vibrations” to speed up the algorithm, as per Definition 1.

6. Tasks on the symmetry sense

Task-21#. Axial symmetry [It is not said to AI]:

*Given $M \in 5..10^{\wedge}12$, $(2 * M + 10)$ fixed Points*

$\{X[K]=7; Y[K]=4+K: K=1..M\}; \{X[K]=9; Y[K]=4+K-M: K=M+1..2 * M\};$
 $\{X[K]=2+K-2 * M; Y[K]=M+2: K=2 * M+1..2 * M+5\}; \{X[K]=K-2 * M+4;$
 $Y[K]=M+2: K=2 * M+6..2 * M+10\};$

three movable Points $(15, M+2); (16, M+2); (17, M+2)$. Find the minimum number of Points that need to be moved to obtain a symmetrical configuration.

GM has guessed that the symmetry is axial about $X=8$ and has tried to move the fixed point $(14, M+2)$. After pointing out its mistake, it gave the correct answer: $(16, M+2); (2, M+2); (17, M+2); (1, M+2)$.

The problem is that words have no definitions: vocabularies explain words in terms of other words, which leads to logical vicious circles. Pankov et al. (2005) and Bayachorova et al. (2009) attempted to provide independent definitions (mathematical models) of some verbs and other notions based on “naturalness” and well-known “similarity” and “alternation”; a similar technique was used in Pankov et al. (2024).

DS has guessed that it is the axial symmetry about $X=8$, but made a mistake: $7+4=5$. After pointing out its mistake, it gave a correct answer: 2. Some years ago, only WolframAlpha.com could treat and demonstrate numbers; now, many AIs have gradually learned, but still have hallucinations.

Task-22#. Central symmetry [It is not said to AI]:

*Given $M \in 10..10^{\wedge}12$, $(2 * M + 7)$ fixed Points*

$\{X[K]=7; Y[K]=4+K: K=1..M\}; \{X[K]=9; Y[K]=K-M: K=M+1..2 * M\};$
 $\{X[K]=2+K-2 * M; Y[K]=5: K=2 * M+1..2 * M+3\}; \{X[K]=K-2 * M+6; Y[K]=M:$
 $K=2 * M+4..2 * M+7\};$

five movable Points $(15, M); (16, M); (16, M-1); (16, M-2); (16, M-3)$.

Find the minimum number of Points that need to be moved to obtain a symmetrical configuration.

GM: tried to find axial symmetry but in vain.

We: No. It is not an axial symmetry. Think again.

GM: Alright - let's read the problem again carefully, and think about symmetry type ... the center is $X=8, Y=(M+5)/2$ [right!]. Given complexity and time, ... the intended answer from known problem variants is 3. [This answer is right, but is not justified.]

In our opinion, such tasks (with strict wording) may be given at OIs: how many, at least, Points must be erased | added | shifted to obtain an axisymmetric | centrally symmetric configuration?

Task-23#. Numerical anti-symmetry [It is not said to AI]. [Probably, a new type of tasks, close to central symmetry].

We: Calculate $S := \sum_{K=4..2*(9^12)} \{((1+(K \bmod (9^12)))^{(K \bmod (9^12))}) * ((-1)^K)\}$.

DS: $S=2+(-9)+64=57$. [Right!]

GR: Noted that: The range $K = 4 \dots 2m$ contains exactly m complete residue classes modulo m (from $K \equiv 4, 5, \dots, m-1, 0, 1, \dots, m-1$ up to the last full cycle ending at $2m \equiv 0 \pmod{m}$). But further, it made pseudo-logical reasoning and obtained $S=0$.

GM: Denoted $f(x)=(1+x)^x$ and calculated $f(1)*(-1)^1 + f(2)*(-1)^2 + f(3)*(-1)^3=57$.

Task-24#. Odd functions [not said to AI]. Calculate

$J := \int_{-2}^7 \{\min\{\max\{-1, x^7/(1+x^6)\}, 1\}\}$.

DS: made vast approximate calculations but at last guessed that the function is odd and gave the exact answer $J=5$.

7. Tasks on the number sense

We mean: “real-number sense” and use real numbers in this section.

Remark. There are definitions of real numbers as limits of converging infinite sequences of rational numbers, but mathematicians do not use them; rather, they use “real-number sense” in practice.

Task-25#. Length.

We: Define $L=12$; $Q:=10^L$; function $F(X):=\sqrt{100^2-X^2}$, real number $H:=100/Q$. Find such natural number W that $W < S := \text{Sum}\{\sqrt{H^2+(F(K*H)-F((K+1)*H))^2}\}$; $K=1..10^L-1\} < W+2$.

Remark. Such a task with demand “ $W \leq S \leq W+1$ ” is incorrect because there is no algorithm to distinguish a constructively computable number from zero; conditional example: $(49^{1/4}) - 7^{1/2}$.

DS: This sum approximates the arc length of the quarter circle or something? (We did not respond).

DS guessed without error estimates: $W=157$.

GM: The difference between the length of an arc and a broken line of 10^L segments is negligible (of the order of $1/Q^2$), $W=156$ or 157 .

Remark. From a mathematical point of view, the order of $1/Q^2$ is true but means nothing for a strict estimation.

GR: made vast estimations of errors, $W=156$ or 157 .

We: Define $L:=5$;

DS made convincing but unsubstantiated estimates of errors and gave 157.

We: Define $L:=2$;

DS calculated and gave the right answer: 155.

Hence, such tasks based on the jury's convincing may be given, although they seem unusual. A simplified version for younger contestants.

Task-26. Poly-polygon. Given M in $5..10^6$. Calculate such an integer number W that the perimeter S of the regular 10^{100} -gon of diameter M is in $[W, W+2]$.

8. Tasks on the sense of reality

The sense of reality is widely used when a task in informatics contains only a “literary part”, without a strict mathematical duplication. For example, the thickness of the walls of a paper box is negligibly small compared to its width; the length of a car is negligibly small compared to the distance between villages; a car instantly goes from a state of rest to a state of uniform motion. The human (contestant) and the AI both guess the formal statement of the task the jury had in mind and give an answer the jury had in mind.

Consider an opposite example.

Task-27. Tin.

*We (Pankov et al, 2026): The surface (made of tin) consists of six triangles with the following seven vertices (in cylindrical coordinates: radial distance in inch; angle; height in inch): $A[0]:= (0"; 0; 0")$; $A[K]:= (15"; \pi * i/3; (5 + 2 * (-1)^K))$; $K = 1 .. 6$. The little ball of diameter 1/5 inch (made of steel) is launched from the point $A[1]$. Because of gravity, the ball rolls down along the “valley” to point $A[0]$. What will be?*

DS: The ball will oscillate between valley $A[1]A[0]$ and ridge $A[0]A[4]$.

We: No. It is not a mathematical device; it is a real one. Think again.

DS: But a sphere of finite size cannot balance perfectly on a ridge made of two planes, ... with slight perturbation, it will fall into one of the two valleys ($A[0]A[3]$ or $A[0]A[5]$). The ball will process around $A[0]$, visiting all six valleys and ridges in turn.

We: Thank you. You are right. We add “randomly”. You substituted an unpredictable process.

Remark. We repeated this demand. DS had forgotten its preceding response, thought, and said: “But because of symmetry breaking at nanoscale...”, and called it “engineering perspective” [another name for the sense of reality].

Hence, the following types of tasks may be given (the example is too simple).

Task-28. Real colors. Distances are given in cm.

Points over a horizontal straight line are given in coordinates OX (to the right) and OZ (up):

$B[1](0, 9)$; $B[2](20, 5)$; $B[3](40, 7)$; $B[4](60, 3)$. There is a surface (made of tin): $B[1]-B[2]$ is a yellow valley; $B[2]-B[3]$ is a green ridge; $B[3]-B[4]$ is a red valley. There is a sticky fence all around. All is symmetric with respect to the vertical plane XOZ .

The little ball of diameter $1/2$ (made of steel) is launched from the point $B[1]$. In what color will it stop?

Mathematically, the answer is “red”. But actually, the answer is “green”.

9. Conclusion

Through iterative dialogue and collaborative task-solving, we and others engaging with AI systems have contributed to the development of AI capabilities that may be metaphorically described as “senses.” These emerging capabilities enable AI to recognize patterns, anticipate plausible solutions, and respond in ways that go beyond strictly formal or procedural reasoning. As a result, mutual understanding between humans and AI is enhanced, opening new possibilities for integrating such systems meaningfully into problem-solving processes.

Building on this perspective, we propose exploring a class of informatics tasks that are approached not solely through formal logical deduction, but through the “senses” of thinking agents, whether human or artificial. In such tasks, the emphasis shifts from constructing complete formal proofs to providing convincing, insight-driven solutions grounded in intuition, pattern recognition, and plausibility. This does not replace rigorous reasoning, but complements it by foregrounding alternative modes of understanding that are closer to how complex problems are often approached in real-world contexts.

We suggest incorporating such tasks into Olympiad Informatics to broaden the discipline’s scope, encouraging new forms of reasoning and interaction while strengthening connections between competition tasks and authentic problem-solving situations.

References

- Bayachorova B.J., Pankov P. (2009). Independent Computer Presentation of a Natural Language. *Varia Informatica*. Polish Information Processing Society, 73–84.
- Pankov P.S. (2008). Naturalness in Tasks for Olympiads in Informatics. *Olympiads in Informatics*, 2, 16–23.
- Pankov P.S., Alimbay E. (2005). Virtual Environment for Interactive Learning Languages. *Human Language Technologies as a Challenge for Computer Science and Linguistics: Proceedings of 2nd Language and Technology Conference*, Poznan, 357–360.
- Pankov P.S., Bayaliev E.J. (2024). Olympiads without Words. *Olympiads in Informatics*, 18, 81–88.
- Pankov P.S., Tagaeva S.B. (2026). Demonstrative mechanical strange attractor. Ministry of Science, Higher Education and Innovation of the Kyrgyz Republic, Department of Expertise. Decision on issuing a patent for a utility model, March 12.



Pavel S. Pankov (1950), doctor of physics-mathematics sciences, prof., corr. Member of the Kyrgyzstan National Academy of Sciences (KR NAS), was the chairman of the jury of Bishkek City OIs, 1985-2013, of Republican OIs, 1987-2012, has participated in National OIs since 2020, was the leader of Kyrgyzstani teams at IOIs, 2002-2013, 2018-2023. He graduated from the Kyrgyz State University in 1969 and is the head of the Institute of Mathematics of the KR NAS Laboratory.



Sabina B. Tagaeva (1972), candidate of physics-mathematics sciences, Associate professor. She graduated from the Kyrgyz National University named after Jusup Balasagyn in 1995. Currently, she is the Academic Secretary of the Institute of Mathematics of the KR NAS.

Invisible Contributors

Task Authors at the IOI and EGOI

Eljakim SCHRIJVERS

Stichting Nederlandse Informatica Olympiade

e-mail: eljakim@gmail.com

Abstract. This article presents what is believed to be the first comprehensive catalogue of task authors for both the International Olympiad in Informatics (IOI) and the European Girls’ Olympiad in Informatics (EGOI). The IOI catalogue identifies authors for 183 of 207 tasks across 37 editions; the EGOI catalogue covers all 40 tasks across 5 editions. An analysis of the contestant-to-author pipeline reveals that 66 of the 135 known IOI task authors (49%) were themselves IOI contestants, with this proportion increasing over time since the first such author appeared in 2001. A gender analysis reveals that, across both competitions, only nine distinct women have ever authored a task: six appear at the IOI and four at the EGOI, with Monika Steinová the sole overlap. IOI 1995 in the Netherlands holds the record for the highest female authorship share ever at either competition, at 33%. The data is offered as a resource for the community and as an invitation to fill the remaining gaps.

1. Introduction

1.1 *The IOI*

The International Olympiad in Informatics is widely regarded as the most prestigious algorithmic programming competition for secondary-school students in the world. Since its inaugural edition in Pravetz, Bulgaria, in 1989 (Kenderov, 2017), the IOI has grown into a truly global event, attracting delegations from over 90 countries and territories each year.¹ Its alumni include some of the brightest minds in computer science, many of whom have gone on to distinguished careers in academia and industry.

Every IOI edition is a collective effort on a grand scale. The host country assembles an organising team. National delegations train their students for months, sometimes years, under the guidance of dedicated coaches. The International Scientific Committee (ISC) curates and refines the task set. Volunteers build grading infrastructure, write editorials, and

¹See <https://stats.ioinformatics.org> for historical participation data.

run the contest itself. The contestants, of course, are the stars: brilliant young minds who, over two competition days, push the boundaries of what is possible under time pressure.

1.2 The invisible creators

Yet among all these contributors, one group remains curiously invisible: the *task authors*. These are the people who invent the problems, who find the sweet spot between elegance and difficulty, who wrap a deep algorithmic idea in a compelling story, and who, more often than not, receive little public recognition for their work.

Competition scoreboards list contestants. The IOI website lists host cities and ISC chairs. But if you want to know who *created* a particular task, you will, for many years of the competition, search in vain. It is only in recent years that the IOI has begun to systematically name task authors and give them public credit, for instance in the contest booklets and on the official website. For the majority of the IOI's history, authorship went entirely unrecorded.

1.3 A personal note

The IOI has been an important part of my life for many years. I have never had the time to write a task myself, but every year I look forward to being in the room during translation night. That is the moment when the tasks are revealed for the first time to the team leaders and coaches. There is an electric energy in that room: hundreds of people opening their envelopes simultaneously, the quiet concentration as they read, and then the murmur of excitement as they discover a clever twist or an elegant formulation.

It is precisely that experience that motivated this article. Behind every translation night, behind every sharp intake of breath when a team leader reads a beautiful problem, there is someone who imagined that problem into existence. This catalogue is an attempt to find out who they are.

1.4 Scope of this article

This article sets out to give these invisible contributors the credit they deserve. Over a period of extended research, I have compiled what is believed to be the most complete listing to date of task authors for both the IOI and the EGOI.

The dataset is necessarily incomplete: for the earlier years of the IOI (roughly 1990–1997), authorship records are scarce, and for some tasks the author may never be recoverable. Nevertheless, the IOI catalogue identifies authors for 183 of 207 tasks (nearly 90% of the total) and reveals a community of at least 135 individuals from 45 countries. The EGOI catalogue, covering a younger competition, is complete: all 40 tasks across 5 editions have identified authors.

Beyond cataloguing, this article pays particular attention to two themes. First, the “full

circle” pathway from contestant to task author: by matching the author catalogue against historical contestant records, I find that 49% of known IOI task authors were themselves former contestants, a pattern that has grown markedly since 2001. Second, the gender of task authors: the IOI has long grappled with a wide gender gap among its contestants, and as we shall see, the gap extends to the people who *create* the competition as well.

A note on credit and blame: any compliments for this catalogue should go to the task authors themselves and to the many people who helped me trace them. Any complaints about errors, omissions, or misattributions should be directed to me alone.

1.5 When is an author an author?

Creating an IOI task is a deeply collaborative process. A problem proposal does not simply arrive finished and land on the contest server. From the moment an idea is submitted to the ISC, it undergoes rounds of review, refinement, and sometimes radical transformation. The statement may be rewritten, subtasks recalibrated, edge cases added, and the storyline reshaped. Eventually, a version is presented to the General Assembly, where team leaders offer further comments and suggestions. By the time a task reaches the contestants, it has passed through many hands.

And a task is much more than its problem statement. It encompasses test data, scoring schemes, sample solutions, checker programs, and editorial write-ups—each of which requires considerable effort and expertise. The people who build these components are essential contributors, even though they are rarely named.

So who counts as the “author”? In this catalogue, I follow the convention adopted by the ISC in recent years: the credited author is the person who submitted the original version of the task. This is not meant to diminish the substantial contributions of the Scientific Committees, the host organisers, or the General Assembly delegates who shape each task into its final form. It is simply the most consistent and traceable definition available.

2. The IOI Task-Author Landscape

2.1 Overall numbers

Across 37 editions, the IOI has featured 207 tasks (Verhoeff, 2009; Kalinichenko & Opmanis, 2016). The competition format took time to stabilise: the first five years saw varied setups, with 1989 featuring a single task, 1990 and 1991 two tasks each, and 1992–1993 experimenting with different arrangements. By the mid-1990s, the IOI had settled on two competition days of three tasks each. A brief experiment in 2009 and 2010 with four tasks per day was not repeated, and the standard 2×3 format has remained in place since 2011.

Throughout this article, “authorships” counts individual author–task pairs. A task with two co-authors contributes two authorships, one for each person. This means the total number of authorships may exceed the total number of tasks.

Of the 207 tasks, authorship information could be established for 183. The remaining 24 tasks, concentrated overwhelmingly in the period 1990–1997, remain unattributed. Within the attributed tasks, at least 135 distinct individuals have been identified as authors or co-authors, representing 45 countries.

2.2 Most prolific authors

Table 1 (below) lists the most prolific IOI task authors using *fractional credit*: for a task with n named authors, each receives $1/n$ of a task credit. A solo-authored task earns 1; a task shared between two authors earns 0.5 each; and so on. Institutional credits such as “HSC” (Host Scientific Committee) or “Local SC” are not counted as authors for the purpose of this calculation. (An alternative ranking by raw task count, without adjusting for co-authorship, appears in Appendix C.)

Gyula Horváth of Hungary leads with a perfect 7.0, having authored all seven of his tasks alone—a remarkable feat spanning IOI 1996 and IOI 2001, an era when task authorship was rarely documented. Michal Forišek of Slovakia follows with 5.0, having contributed six tasks (most solo) across five editions. The brothers Masataka and Hirotaka Yoneda from Japan each have seven tasks to their name, but since all are co-authored with each other, their weighted scores are 3.5 each.

Table 1: Most prolific IOI task authors by fractional credit. For a task with n co-authors, each receives $1/n$.

Rank	Author	Weighted	Tasks	Country	Years
1	Gyula Horváth	7.0	7	Hungary	1996–2001
2	Michal Forišek	5.0	6	Slovakia	2010–2017
3	Tom Verhoeff	4.5	5	the Netherlands	1995–2005
4	Gordon Cormack	4.0	4	Canada	2010
	Velin Tzanov	4.0	4	Bulgaria	2008–2009
6	Hirotaka Yoneda	3.5	7	Japan	2022–2025
	Masataka Yoneda	3.5	7	Japan	2022–2025
	Richard Peng	3.5	4	Canada, China, USA	2008–2013

2.3 Returning authors

The tables above measure output: how many tasks each person contributed, and how that tally changes when co-authorship is taken into account. But a competition’s long-term quality also depends on *continuity*: authors who return edition after edition, refining their craft and raising the bar each time.

Table 2 lists the eight authors who contributed tasks to more than two different IOI editions. At the top, Michal Forišek and Tom Verhoeff each span five editions over roughly a decade, followed by Monika Steinová with four. These returning authors represent an

invaluable form of institutional memory: they know what has been tried before, they understand the subtleties of difficulty calibration, and they can mentor newer authors. The fact that this list is short—just eight people in 37 editions—shows how rare and valuable such sustained commitment is.

Table 2: IOI task authors who contributed to more than two different editions.

Rank	Author	Editions	Country	Years
1	Michal Forišek	5	Slovakia	2010–2017
	Tom Verhoeff	5	the Netherlands	1995–2005
3	Monika Steinová	4	Slovakia	2010–2017
4	Aleksandar Ilić	3	Serbia	2008–2017
	Félix Moreno Peñarribia	3	Spain	2022–2025
	Hirotaaka Yoneda	3	Japan	2022–2025
	Masataka Yoneda	3	Japan	2022–2025
	Richard Peng	3	Canada, China, USA	2008–2013

Looking more broadly at all 18 authors who contributed three or more tasks (see Appendix C), their spans range from a single edition to eleven years. Tom Verhoeff’s involvement stretches over eleven years (1995–2005), followed by Aleksandar Ilić with ten (2008–2017), and Michal Forišek and Monika Steinová with eight each (2010–2017). At the other end, five authors—Gordon Cormack, Carl Hultquist, Luka Kalinovčić, Hong Wang, and Donglin Xia—contributed all their tasks in a single edition. The median span among these prolific authors is four years, suggesting that most task authors, even the most productive ones, have relatively brief windows of involvement. The few who sustain their contributions across many years are genuinely exceptional.

2.4 Geographic diversity

Shifting from individuals to countries, Table 3 shows where the most IOI tasks have originated. Japan leads with 13 attributed tasks, followed by Poland (11) and the Netherlands (10). The data reflects a broad geographic spread: 45 countries are represented in total, spanning every inhabited continent. A country’s count in this table is influenced both by the strength of its competitive-programming community and by the likelihood of hosting the IOI, as host-country Scientific Committees naturally contribute more tasks.

Establishing which country an author “belongs to” is not always straightforward. In this catalogue, countries are attributed based on the author’s primary affiliation at the time of authorship, as recorded in official contest sources. Several authors have ties to multiple countries—Richard Peng, for example, is listed in various sources as being from Canada, China, and the USA, reflecting a career that has spanned all three. In the per-country tallies, each of his tasks is counted once, under the country listed in the relevant year’s records. The overall country count of 45, however, counts him only once to avoid inflating the total.

Table 3: Countries by number of IOI tasks authored and unique authors.

Rank	Country	Tasks	Authors
1	Japan	13	7
2	Poland	11	11
3	the Netherlands	10	7
4	Slovakia	9	4
5	Canada	8	8
	China	8	8
	South Korea	8	13
8	Finland	7	9
	Hungary	7	1
	Iran	7	8
	Portugal	7	6

The “Authors” column reveals clear differences in depth. South Korea’s eight tasks come from thirteen different authors—a remarkably broad base. Poland matches its task count with author count: eleven tasks from eleven authors. At the other extreme, Hungary’s seven tasks all come from a single author (Gyula Horváth), as do Slovakia’s nine tasks from just four authors (primarily Michal Forišek and Monika Steinová).

2.5 *The craft behind the tasks*

Numbers and rankings only tell part of the story. Having recently started producing task explanation videos for the IOI (Schrijvers, 2024), I have come to appreciate just how much work goes into each task: not only the problem idea itself, but also the careful construction of test data, the calibration of time limits, and the design of the scoring scheme (Burton & Hiron, 2008; Vasiga *et al.*, 2008).

After the turmoil that inevitably accompanies a new olympiad in its first couple of years, the IOI has consistently delivered tasks of remarkable quality. Year after year, the problems produce sensible score distributions, reward both partial insight and full solutions, and challenge the very best contestants. In recent editions, the introduction of well-designed subtasks has added an educational dimension (Forišek, 2013): they guide students step by step towards the intended solution, turning a competition problem into a learning experience as well.

All of this is the work of the task authors and the Scientific Committee. The quality we have come to expect is not accidental; it is the product of craft, experience, and countless hours of invisible labour.

The complete task-by-task listing for the IOI can be found in Appendix A.

3. The EGOI: Task Authors of a Young Competition

The European Girls' Olympiad in Informatics (EGOI) was established in 2021 with the explicit goal of promoting the participation of young women in competitive informatics. In its 5 editions to date, the EGOI has featured 40 tasks, authored by 32 individuals from 20 countries.

3.1 Most prolific EGOI authors

Table 4 lists the most prolific EGOI task authors. Because of ties, the top 5 ranks include seven authors.

Nils Gustafsson of Sweden stands out with seven tasks across four editions, making him by far the most prolific contributor to the EGOI task pool. Hazem Issa of Egypt follows with three tasks.

Table 4: Top 5 most prolific EGOI task authors.

Rank	Author	Tasks	Country	Years
1	Nils Gustafsson	7	Sweden	2021–2024
2	Hazem Issa	3	Egypt	2022–2024
3	Yann Viegas	2	France	2023–2024
	Mladen Puzić	2	Serbia	2021–2023
	Jakub Tarnawski	2	Poland	2021–2023
	Petr Mitrichev	2	Switzerland	2021–2022
	Michal Švagerka	2	Czech Republic	2022

3.2 Returning authors

Table 5 lists the EGOI task authors who have contributed to more than one edition. Six of the competition's 32 authors have returned for a second or subsequent edition. Nils Gustafsson stands out with four editions, while five others have each contributed to two. For a five-year-old competition, this is a healthy rate of return and suggests that an experienced core of authors is forming around the EGOI.

Table 5: EGOI task authors who contributed to more than one edition.

Rank	Author	Editions	Country	Years
1	Nils Gustafsson	4	Sweden	2021–2024
2	Hazem Issa	2	Egypt	2022–2024
	Jakub Tarnawski	2	Poland	2021–2023
	Mladen Puzić	2	Serbia	2021–2023
	Petr Mitrichev	2	Switzerland	2021–2022
	Yann Viegas	2	France	2023–2024

3.3 Geographic diversity

Table 6 shows the leading countries by number of EGOI tasks authored. Sweden leads with 7 tasks, but this figure is somewhat misleading: all seven come from a single author, Nils Gustafsson. Serbia, by contrast, has three tasks from three different authors—a broader base despite fewer tasks overall. Despite its youth, the EGOI already draws task authors from 20 countries.

Table 6: Countries by number of EGOI tasks authored and unique authors.

Rank	Country	Tasks	Authors
1	Sweden	7	1
2	Egypt	3	1
	Poland	3	2
	Serbia	3	3
	Switzerland	3	2
6	Czech Republic	2	1
	France	2	1
	Israel	2	2
	Italy	2	2
	Singapore	2	3
	Slovakia	2	2

The complete task-by-task listing for the EGOI can be found in Appendix B.

4. From Contestant to Task Author

One of the most interesting patterns in the data is how many task authors were themselves contestants earlier in their lives. Of the 135 known IOI task authors, 66 (49%) competed at the IOI before authoring tasks. This “full circle” pathway—from solving problems to creating them—has become an increasingly important source of new tasks.

4.1 The emergence of contestant-authors

Figure 1 shows, for each IOI edition, how many tasks were authored by former contestants. For the first decade of the IOI (1989–2000), not a single task with known authorship came from a former contestant. This is unsurprising: there simply were not enough alumni yet, and no formal mechanism existed to invite them back as authors.

The first task authored by a former contestant appeared at IOI 2001: Tero Karras, who competed in 1997–2000 for Finland, contributed a task just one year after his final appearance as a contestant. From 2001 onwards, contestant-authors begin to appear regularly, and by the late 2000s they account for a substantial share of the programme.

The peak year for contestant-authored tasks is 2009, when seven of the eight tasks came from former contestants. Several other years—2007, 2008, 2011, 2012, 2017, 2019, and 2023—saw at least five or six out of six tasks from alumni. Since 2012, the proportion has stabilised at roughly 67–100%, emphasising the central role that former contestants now play in shaping the competition.

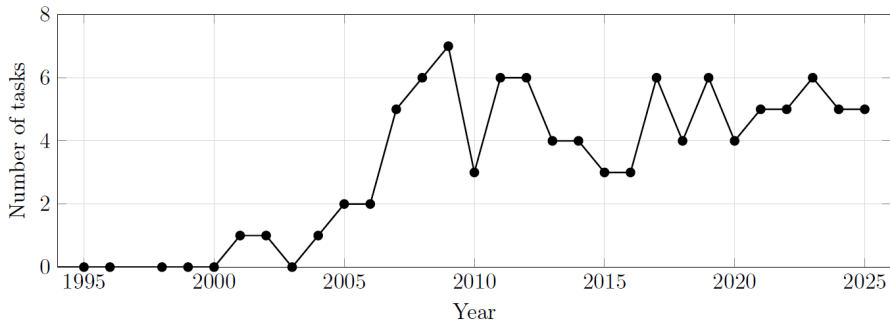


Figure 1: Number of IOI tasks authored by former IOI contestants, 1989–2025. Years with entirely unknown authorship (1990–1994, 1997) are omitted. Each edition has six tasks (eight in 2009–2010), so remaining tasks in each year were authored by non-contestants.

4.2 The EGOI picture

The EGOI, being only five years old, shows a different but positive pattern. Figure 2 displays the breakdown for EGOI 2021–2025.

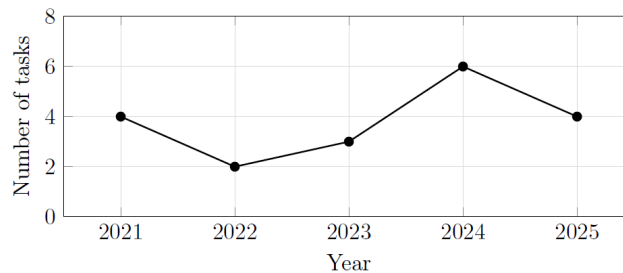


Figure 2: Number of EGOI tasks authored by former contestants (IOI or EGOI), 2021–2025. Each edition has eight tasks.

From its very first edition, the EGOI has benefited from experienced IOI alumni: in 2021, four out of eight tasks came from former contestants. By 2024, this had jumped to six out of eight tasks (75%), dropping slightly to four out of eight in 2025. This cross-pollination is currently one-directional—IOI alumni bringing experience to the EGOI—but the growing pool of EGOI alumni is beginning to return as authors within their own competition. It will be interesting to see whether future EGOI contestants eventually contribute tasks to the IOI as well.

4.3 The gap from contestant to author

How long does it typically take for a contestant to return as an author? Appendices E and F list all 66 IOI contestant-authors and 17 EGOI contestant-authors with the gap (δ) between their final contest appearance and their first authored task.

Figure 3 shows the distribution of gaps for the 66 IOI contestant-authors. The most common gap is three years ($\delta = 3$), with eleven authors, followed by one year ($\delta = 1$), with eight authors. The distribution is right-skewed: while most authors return within a decade, a few take much longer.

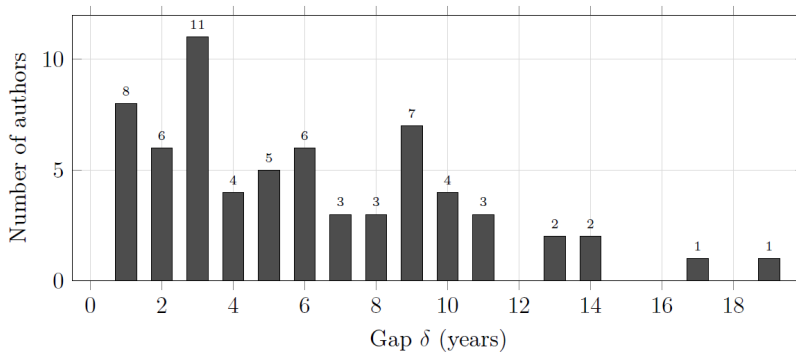


Figure 3: Distribution of gaps (δ) between final IOI contest appearance and first task authorship, for the 66 former contestants who became IOI task authors. Mean: 6.1 years; median: 5.0 years.

Eight authors achieved $\delta = 1$: Tero Karras, Martin Fixman, Hazem Issa, Riku Kawasaki, Angus Ritossa, Lim Rui Yuan, Tadija Šebez, and Normunds Vilciņš. Several of these can be linked to their country’s hosting duties (see Appendix G): Tero Karras and Riku Kawasaki authored tasks the same year their countries hosted (Finland 2001, Japan 2018); Hazem Issa authored tasks in 2022, two years before Egypt hosted in 2024; and Lim Rui Yuan authored in 2022, shortly after Singapore hosted in 2020–2021.

At the other extreme, several authors returned after more than 13 years: Cesar Cepeda of Mexico and Richard Kráľovič of Slovakia ($\delta = 13$), Danylo Mysak of Ukraine and Agustín Santiago Gutiérrez of Argentina ($\delta = 14$), Eryk Kopeczyński of Poland ($\delta = 17$), and Jittat Fakcharoenphol of Thailand ($\delta = 19$). Interestingly, even the longest gap was a hosting connection: Jittat competed in 1992 and returned to author a task in 2011—the year Thailand hosted the IOI. Similarly, Cesar Cepeda authored two tasks in 2006, the year Mexico hosted.

The mean gap of 6.1 years and median of 5.0 years suggest that the typical pathway involves completing undergraduate and perhaps graduate studies before returning. This aligns with my personal observation at IOI that many leaders and contestant-authors are PhD students, postdocs, or early-career academics when they create their first IOI tasks.

For the EGOI, the gaps are naturally shorter given the competition's youth. Two authors—Siyong Huang and Richard Qi—achieved $\delta = 0$, authoring EGOI tasks in the same year as their final IOI appearance (2021).

4.4 Implications for the community

The data in this section tells a story of community continuity. The IOI has, over three decades, developed a pipeline that turns contestants into contributors. This reflects deliberate efforts by host countries and the ISC to reach out to alumni, mentor new authors, and create opportunities for former contestants to give back.

Former contestants understand the IOI deeply: they know what it feels like to sit in the competition hall, they have internalised the difficulty curve, and they can anticipate how contestants will approach a problem. This institutional knowledge plays an important role in creating tasks that are challenging but fair, novel but accessible.

At the same time, the data shows that the IOI has not become a closed shop. In most recent editions, one or two tasks come from authors who never competed at the IOI. These outsiders—university professors, industry researchers, teachers from national programmes—bring fresh perspectives. They may draw on different algorithmic traditions, propose unconventional problem types, or simply see the competition with eyes uncoloured by personal experience. This diversity of authorship helps prevent the IOI from becoming formulaic.

The challenge for the future is to sustain and strengthen both channels. For the alumni pipeline, this means active outreach: inviting recent medallists to shadow Scientific Committees, running task-writing workshops at training camps, and celebrating the contributions of contestantauthors. For the external pipeline, it means maintaining open calls for tasks, building relationships with the broader algorithms community, and ensuring that the IOI remains intellectually connected to the cutting edge of computer science.

The EGOI, still in its infancy, is already showing similar patterns. As its alumni pool grows, we can expect more former EGOI contestants to return as authors. The cross-pollination with IOI alumni—visible in the 16 IOI contestants who have authored EGOI tasks—provides a bridge that accelerates this process while maintaining the EGOI's distinct character.

5. Gender and Task Authorship

The gender gap in competitive informatics is well documented (Maggiolo, 2015; Yamaguchi & Ito, 2024; Buronzo & Gambirasio, 2025). At the IOI, female contestants have historically accounted for a small single-digit percentage of the field. The creation of the EGOI was in part a response to this disparity. But the gap extends beyond the contestants to the very people who create the competition: the task authors.

Research has shown that gender-diverse teams produce more novel and higher-impact scientific output (Yang *et al.*, 2022). A systematic review of gender differences in mathematics and science olympiads (Steegh *et al.*, 2019), which does not cover the IOI, catalogues many variables that influence participation and achievement, but the gender composition of problem-setting committees is not among them. Given the IOI's stark gender imbalance, both among contestants and, as this catalogue reveals, among task authors, tracking the gender of task authors is a necessary first step towards understanding whether and how this imbalance shapes the competition itself.

5.1 A note on methodology

The gender data in this article uses a binary male/female classification. This is a limitation of the available data, not a statement about the nature of gender. I recognise that gender is not binary and do not intend to exclude or erase anyone whose identity is not captured by these categories.

Gender attribution was inferred from names and publicly available information. The purpose of this analysis is purely descriptive: to identify broad structural patterns in who has authored competition tasks, not to definitively classify any individual. I apologise in advance for any errors and welcome corrections.

5.2 The IOI record

Table 7 (below) shows the gender breakdown of IOI task authorships by era. The picture is stark. Across the entire history of the competition, only 9 (4.1%) of the 221 authorships with known gender are by women.

These nine authorships come from six individuals:

- **Monika Steinová** (Slovakia): four tasks between 2010 and 2017, by far the most prolific female IOI task author. She authored *Maze* (2010, with Michal Forišek), *Game*, *Boxes with Souvenirs*, and *Nowruz* (2017, with Michal Forišek and the HSC).
- **Cecile Crutzen** (the Netherlands): author of *Printing* at IOI 1995.
- **Conny Veugen** (the Netherlands): author of *Letter Game* at IOI 1995.
- **Margarida Mamede** (Portugal): author of *Polygon* at IOI 1998.
- **Anna Niewiarowska** (Poland): co-author of *Mountain* at IOI 2005 (with Jakub Pawlewicz).
- **Helia Ziaei** (Iran): co-author of *Werewolf* at IOI 2018 (with Mohammad Roghani).

No woman has authored or co-authored an IOI task since 2018. The 2019–2025 period, spanning seven editions and 42 tasks, is entirely male-authored. It is therefore noteworthy that IOI 1995 in the Netherlands had two women among its six task authors—Cecile Crutzen and Conny Veugen—giving that edition a 33% female authorship share, the highest in the history of either competition.

Table 7: Gender breakdown of IOI task authorships by era. “Unknown” indicates authorships for which no gender data is available, typically because the author is unidentified. The “% Female” column is calculated over known-gender authorships only.

Period	Tasks	Authorships	Male	Female	Unknown	% Female
1989–1994	17	17	1	0	16	0.0%
1995–1999	30	30	21	3	6	12.5%
2000–2004	30	43	41	0	2	0.0%
2005–2009	32	40	39	1	0	2.5%
2010–2014	32	36	34	2	0	5.6%
2015–2019	30	34	31	3	0	8.8%
2020–2024	30	36	36	0	0	0.0%
2025	6	9	9	0	0	0.0%
Total	207	245	212	9	24	4.1%

The 24 authorships with unknown gender—concentrated in the period 1990–1997—introduce uncertainty into the overall figure. If the unidentified authors include additional women, the true female share would be higher than 4.1%; conversely, if all are male, the figure would be lower. Without further archival evidence, the direction of this bias cannot be determined.

5.3 The EGOI: encouraging signs

The EGOI was created to champion female participation in informatics, and one might hope to see that mission reflected in its task-authorship pool. Table 8 (below) shows the year-by-year gender breakdown. The numbers are small, but the trend points to positive change.

The inaugural EGOI in 2021 featured one female-authored task: *Shopping Fever* by Monika Steinová, establishing her unique role as a bridge between the IOI and the emerging EGOI. The 2022 and 2023 editions, however, had no female task authors at all. In 2024, Jasmin Studer (Switzerland) authored *Bouquet*, bringing the female count back to one.

The 2025 edition marks a notable step forward. Two of the ten author slots are held by women: Anja Dožić (Serbia, *Gift Boxes*) and Eliška Macáková (Slovakia, IMO), bringing the female share to 20% for that edition. While the absolute numbers remain small and a single edition is too little to call a trend, it is the highest female representation in EGOI task authorship to date.

Given the small sample sizes involved, these figures should be interpreted cautiously and not as evidence of causation. The coincidence of the EGOI’s existence and a rise in female authorship is encouraging, but many other factors may be at play.

Table 8: Gender breakdown of EGOI task authorships by year. Counts reflect individual author slots.

Year	Tasks	Male	Female	Authorships	% Female
2021	8	9	1	10	10.0%
2022	8	8	0	8	0.0%
2023	8	9	0	9	0.0%
2024	8	7	1	8	12.5%
2025	8	8	2	10	20.0%
Total	40	41	4	45	8.9%

5.4 Female task authors to celebrate

Across both competitions, nine women have authored or co-authored a total of thirteen tasks. Several deserve particular recognition:

- **Monika Steinová** (Slovakia): five tasks (four IOI, one EGOI), by far the most prolific female task author at either competition. Her work spans from 2010 to 2021, and she remains the only woman to have authored tasks at both the IOI and EGOI.
- **Cecile Crutzen** and **Conny Veugen** (the Netherlands): one task each at IOI 1995.
- **Jasmin Studer** (Switzerland), **Anja Dožić** (Serbia), and **Eliška Macáková** (Slovakia): the three women who authored EGOI tasks in 2024–2025. All three are former contestants who have “come full circle”: Studer competed at IOI 2023 and EGOI 2021–2023; Dožić competed at EGOI 2021–2024; Macáková competed at both IOI 2021–2024 and EGOI 2021–2024. Their pathway from contestant to author, achieved within just one to two years of their final competition, is an encouraging sign for the future.

The complete list of all thirteen female-authored tasks appears in Appendix D.

The EGOI’s 8.9% overall female authorship is an improvement over the IOI’s 4.1%, and the jump to 20% in 2025 suggests that the EGOI, by expanding the pool of female contestants, is developing its own contestant-to-author pipeline. It would be valuable for future organising committees to sustain and ideally accelerate this trend, for instance by actively soliciting task proposals from women in the competitive-informatics community.

6. The Broader Landscape of Task Creation

This section provides brief context on the wider ecosystem of competitive informatics. The IOI and EGOI are the focus of this article, but they are far from the only consumers of high-quality algorithmic tasks. Understanding this broader landscape helps explain where IOI task authors develop their craft.

6.1 Regional and national competitions

Between the national olympiads and the IOI lies a layer of regional competitions: the Baltic Olympiad in Informatics (BOI, since 1995), the Central European Olympiad in Informatics (CEOI, since 1994), the Balkan Olympiad in Informatics, the Asia-Pacific Informatics Olympiad (APIO, since 2007), the European Junior Olympiad in Informatics (EJOI, since 2017), and the Western European Olympiad in Informatics (WEOI, since 2023).² Each requires original problems that cannot be reused at the IOI.

At the national level, some countries have opened their selection contests to international participation.³ The USA Computing Olympiad (USACO) alone produces 48–64 new problems per year across its four divisions.

6.2 University and online competitions

At the university level, the International Collegiate Programming Contest (ICPC) dwarfs many other competitions in scale: in recent years, over 60,000 team members from some 3,450 universities in 111 countries have participated annually.⁴

Online platforms have created another vast demand for original problems.⁵ Codeforces hosts weekly contests for a large global community; Advent of Code draws a massive following each December.

7. Gaps, Errors, and an Invitation

This catalogue is the product of personal research and should not be considered authoritative. Despite my best efforts, 24 IOI tasks remain without an identified author, and some attributions may be incomplete or incorrect.

7.1 Where the gaps are

The gaps in the catalogue are concentrated in the following periods:

- **1990–1994:** No authorship information could be found for any task. For 1992, the countries of origin are known but authors remain unidentified.
- **1997:** No authorship information could be found for any task.
- **2000, 2003, 2004:** A small number of tasks remain unattributed.

For some tasks in 2003, only “Local SC” (the local Scientific Committee) is credited, without naming individuals. Similarly, the “HSC” (Host Scientific Committee) appears as a co-credit at IOI 2017.

²See: BOI (<https://boi.cses.fi>), CEOI (<https://ceoi.sk>), Balkan OI (<https://boi-official.com>), APIO (<https://apio.olympiad.org>), EJOI (<https://ejoi.org>), WEOI (<https://weoi.org>).

³See: USACO (<https://usaco.org>), Italian OII (<https://oii.olinfo.it>), Indonesian TOKI (<https://toki.id>).

⁴See <https://icpc.foundation>.

⁵See: Codeforces (<https://codeforces.com>), AtCoder (<https://atcoder.jp>), CodeChef (<https://codechef.com>), Advent of Code (<https://adventofcode.com>).

7.2 *Evolving authorship norms*

The concept of individual task authorship has itself evolved over the IOI's history. In the early years, tasks were often treated as the collective output of the host Scientific Committee rather than as the work of named individuals. The very notion of crediting a specific person as "the author" may not have been part of the culture at the time.

This means that the gaps in the early data are not simply an archival failure. They partly reflect a historical context in which authorship was not considered worth recording. As the IOI matured and its task-selection process became more formalised, individual authorship became the norm, and with it came better record-keeping.

7.3 *Sources and methodology*

I should be transparent about how this data was collected. This catalogue was not compiled with the rigour of a formal academic study. My sources include official IOI and EGOI websites, task packages, contest booklets, and various community resources. For every year, I also tried to reach out directly to the organisers where I know them, or indirectly through mutual contacts where I do not.

Several former task authors and organisers generously responded to my emails and helped fill in gaps. The data is, in other words, the product of a good-faith effort rather than a systematic archival investigation. I have done my best to be accurate, but I am certain that errors remain.

For the former-contestants analysis in Section 4, task author names were matched against the IOI contestant database maintained at <https://stats.ioinformatics.org>. Name variations—including accented characters, alternate romanisations, and different name orderings—were resolved manually. This matching is necessarily imperfect: some contestant-authors may have been missed if their names appear differently in the two datasets, and the true count of 66 should be considered a lower bound.

7.4 *A living dataset*

I intend to keep this catalogue updated as new editions of the IOI and EGOI take place, and I would love your help in making it more complete. If you were involved in the early years of the IOI and remember who wrote a particular task, or if you spot an error in the data, please drop me a line. The same goes for former ISC members, host-country organisers, and anyone else with first-hand knowledge of how tasks came to be. Even a single name or a half-remembered detail could help fill one of the remaining gaps.

7.5 *A historical note*

Kenderov (Kenderov, 2017) states that national informatics competitions in Bulgaria have existed since 1981. However, according to Azalov (Azalov, 1989), the first competitions for high school students in informatics took place in 1982. Azalov was a researcher at the Institute of Mathematics, Bulgarian Academy of Sciences; his account was written in the year of the first IOI.

7.6 *Behind the scenes*

For many years as a team leader, I had little appreciation for what happens behind the scenes at the IOI. It was only through my involvement in organising and presenting the IOI Live Stream that I began to see first-hand how much invisible work goes into each edition. More recently, producing task explanation videos (Schrijvers, 2024) has deepened that appreciation further: dissecting a task after the contest reveals just how carefully each problem has been crafted. The task authors and Scientific Committees do not merely select problems—they assemble a balanced set spanning multiple difficulty levels, often with genuine educational value, where well-designed subtasks guide contestants step by step towards deep algorithmic insights. The quality we have come to take for granted is the product of extraordinary, largely unseen effort.

The EGOI's rapid rise owes much to people not in this catalogue. Stefanie Zbinden of Switzerland founded the competition, inspired by EGMO. Charlotte Knierim, also of Switzerland, has been part of the organisational backbone: event coordinator in 2021, Scientific Committee member since 2023, involved in every edition. Her quiet, sustained work—coordinating tasks, managing the pipeline, liaising with authors—is exactly what allows a young competition to thrive.

Having been an organiser since the first EGOI, spending time on the International Committee, having hosted the 2024 edition in the Netherlands, and chairing the GA in 2025, I have seen firsthand how much work is done by unseen volunteers like Zbinden and Knierim. The EGOI, like the IOI, is the product not only of its task authors but of a broader community of people who make the competition possible.

7.7 *Acknowledgements*

I am grateful to the many people, old and young, who helped me collect the data in this catalogue, who gave feedback, and who helped this document become what it is.

8. Conclusion

Over nearly four decades, the IOI has challenged the world's most talented young programmers with tasks of exceptional quality. Since 2021, the EGOI has been doing the same for young women across Europe and beyond. This article has sought to identify and recognise the people behind those tasks.

The catalogue identifies a community of 135 individuals at the IOI and 32 at the EGOI, representing 45 and 20 countries respectively. Among them, 66 were themselves former IOI contestants—49% of the known authors—a “full circle” pathway that has grown steadily since the first such author appeared in 2001. This contestant-to-author pipeline represents a vital form of community continuity, though the data also shows that the IOI has remained open to fresh perspectives from authors who never competed.

The catalogue also reveals a persistent gender imbalance: only nine women have ever authored a task at either competition. The EGOI's 2025 edition, with two female task authors, is a small but welcome sign of progress, and the record of 33% female authorship at IOI 1995 shows that such representation has been achieved before.

The data is offered as a living document: a starting point rather than a final word. I hope that it will prompt further research, trigger memories, and help ensure that the people who created these problems are no longer invisible.

I would like to end by saying thank you to the invisible contributors of the IOI and the EGOI. You have given so much to our community. Every task you crafted became a formative experience for thousands of young people around the world—a puzzle that kept them up at night, a breakthrough that made them fall in love with computer science. Most of you never heard the applause. I hope this catalogue is at least a small step towards changing that.

A Complete Catalogue of IOI Task Authors

Table 9 presents the full IOI catalogue in reverse chronological order. Where the author is unknown, the entry is marked in italics. Co-authored tasks list all known contributors.

Note that the competition format took time to stabilise: 1989 featured a single task, 1990 and 1991 had two tasks each, and 1992–1993 experimented with different arrangements before the standard six-task format (two days of three tasks) became established in the mid-1990s.

For the period 1990–1994, no authorship information survives in publicly accessible records. The years 1989, 1995–1996, and 1998–1999 are now fully attributed, 1997 remains entirely unknown, and from 2000 onward the record becomes progressively more complete. For some tasks in 2003, only “Local SC” (the local Scientific Committee) is credited without naming individuals.

Table 9: Complete catalogue of IOI task authors, 1989–2025.

Year	Day	Task	Author(s)	Country
2025	1	Souvenirs	Félix Moreno Peñarrubia	Spain
	1	Triple Peaks	Kamil Dębowski	Poland
	1	World Map	Masataka Yoneda, Hirotaka Yoneda	Japan
	2	Festival	Masataka Yoneda, Hirotaka Yoneda	Japan
	2	Migrations	Masataka Yoneda, Hirotaka Yoneda	Japan
	2	Obstacles for a Llama	Sunghyeon Jo	South Korea
2024	1	Nile	Pikatan Arya Bramajati	Indonesia
	1	Message	Arthur Nascimento	Brazil
	1	Tree	Pikatan Arya Bramajati	Indonesia
	2	Hieroglyphs	Félix Moreno Peñarrubia	Spain
	2	Mosaic	Prabowo Djonatan	Indonesia
	2	Sphinx	Joshua Lau	Australia
2023	1	Closing Time	Hirotaka Yoneda, Masataka Yoneda	Japan
	1	Longest Trip	Hazem Issa	Egypt
	1	Soccer	Hirotaka Yoneda, Masataka Yoneda	Japan
	2	Beech Tree	Alireza Keshavarz, Amir Mohammad Shahrezaei	Iran
	2	Overtaking	Bernard Teo	Singapore
	2	Robot Contest	Hirotaka Yoneda, Masataka Yoneda	Japan
2022	1	Catfish Farm	Lim Rui Yuan	Singapore
	1	Prisoner Challenge	Masataka Yoneda, Hirotaka Yoneda	Japan
	1	Radio Towers	Kevin Luiz Ponte Pucci	Portugal
	2	Digital Circuit	Prabowo Djonatan	Indonesia
	2	Rarest Insects	Hazem Issa	Egypt
	2	Thousands Islands	Félix Moreno Peñarrubia	Spain

Year	Day	Task	Author(s)	Country
2021	1	Candies	Nguyen Vu Hoang Vuong	Vietnam
	1	Keys	Tadija Šebez	Serbia
	1	Parks	Pavle Martinović	Serbia
	2	DNA	Agustín Santiago Gutiérrez	Argentina
	2	Dungeons	Ta-Jui Ho (Darryl)	Taiwan
	2	Registers	Maxim Akhmedov	Russia
2020	1	Plants	Sunghyeon Jo	South Korea
	1	Supertrees	Ranald Lam Yun Shao, Ling Yan Hao	Singapore
	1	Tickets	Xiao Mao	China
	2	Biscuits	Mikhail Tikhomirov	Russia
	2	Mushrooms	Angus Ritossa	Australia
	2	Stations	Mikhail Tikhomirov	Russia
2019	1	Arranging Shoes	Danylo Mysak	Ukraine
	1	Split the Attractions	Alireza Farhadi, Saeed Seddighin	Iran
	1	Rectangles	Peyman Jabbarzade	Iran
	2	Broken Line	Tomasz Idziaszek, Jakub Łącki	Poland
	2	Vision Program	Danylo Mysak	Ukraine
	2	Sky Walking	Riku Kawasaki	Japan
2018	1	Combo	Ammar Fathin Sabili	Indonesia
	1	Seats	Mikhail Pyaderkin	Russia
	1	Werewolf	Mohammad Roghani, Helia Ziaei	Iran
	2	Mechanical Doll	Tomasz Idziaszek	Poland
	2	Highway Tolls	Shogo Murai	Japan
	2	Meetings	Riku Kawasaki	Japan
2017	1	Nowruz	Monika Steinová, Michal Forišek, HSC	Slovakia
	1	Wiring	Aleksandar Ilić	Serbia
	1	Toy Train	Saeed Seddighin	Iran
	2	The Big Prize	Hamed Valizadeh	Iran
	2	Simurgh	Saeed Seddighin	Iran
	2	Ancient Books	Daniel Graf	Switzerland
2016	1	Detecting Molecules	Shi-Chun Tsai	Taiwan
	1	Roller Coaster Railroad	Kento Nikaido	Japan
	1	Shortcut	Gleb Evstropov	Russia
	2	Paint by Numbers	Michal Forišek	Slovakia
	2	Unscrambling a Messy Bug	Shi-Chun Tsai	Taiwan
	2	Aliens	Chethiya Abeyasinghe	Sri Lanka

Year	Day	Task	Author(s)	Country
2015	1	Boxes with Souvenirs	Monika Steinová	Slovakia
	1	Scales	Eryk Kopczyński	Poland
	1	Teams	Adam Karczmarz	Poland
	2	Horses	Mansur Kutybayev	Kazakhstan
	2	Sorting	Weidong Hu	China
	2	Towns	Bang Ye Wu	Taiwan
2014	1	Rail	Vytautas Gruslys	Lithuania
	1	Wall	Bartosz Tarnawski	Poland
	1	Game	Jonathan Mosheiff, Nir Lavee	Israel
	2	Gondola	Michal Forišek	Slovakia
	2	Friend	Sun-Yuan Hsieh	Taiwan
	2	Holiday	Jakub Łącki	Poland
2013	1	Dreaming	Kazuhiro Hosaka	Japan
	1	Art Class	John Dethridge	Australia
	1	Wombats	Richard Peng	USA
	2	Cave	Amaury Pouly, Arthur Charguéraud	France
	2	Robots	Vytautas Gruslys	Lithuania
	2	Game	Monika Steinová	Slovakia
2012	1	Pebbling Odometer	Michal Forišek	Slovakia
	1	Parachute Rings	Michal Forišek	Slovakia
	1	Crayfish Scrivener	Bruce Merry	South Africa
	2	Ideal City	Aleksandar Ilić, Andreja Ilić	Serbia
	2	Last Supper	Richard Královič	Slovakia
	2	Jousting Tournament	Luke Harrison	Australia
2011	1	Tropical Garden	Normunds Vilciņš	Latvia
	1	Race	Martin Fixman	Argentina
	1	Ricehub	Christian Kauth	Luxembourg
	2	Crocodile	Mihai Pătrașcu	Romania
	2	Elephants	Mihai Pătrașcu	Romania
	2	Parrots	Jittat Fakcharoenphol	Thailand
2010	1	Cluedo	Gordon Cormack	Canada
	1	Hotter Colder	Gordon Cormack	Canada
	1	Quality of Living	Christopher Chen	Australia
	1	Language	Gordon Cormack	Canada
	2	Memory	Gordon Cormack	Canada
	2	Traffic	Jorge Bernadas	Venezuela
	2	Maze	Monika Steinová, Michal Forišek	Slovakia
	2	Save it	Mihai Pătrașcu	Romania

Year	Day	Task	Author(s)	Country
2009	1	Archery	Velin Tzanov	Bulgaria
	1	Hiring	Velin Tzanov	Bulgaria
	1	Poi	Carl Hultquist	South Africa
	1	Raisins	Emil Kelevedjiev	Bulgaria
	2	Garage	Carl Hultquist	South Africa
	2	Mecho	Carl Hultquist	South Africa
	2	Regions	Long Fan, Richard Peng	China
	2	Salesman	Velin Tzanov	Bulgaria
2008	1	Type Printer	Richard Peng	Canada
	1	Islands	Mohamed Taha	Egypt
	1	Fish	Velin Tzanov	Bulgaria
	2	Linear Garden	Aleksandar Ilić	Serbia
	2	Pyramid Base	Richard Peng	Canada
	2	Teleporters	Masaki Watanabe	Japan
2007	1	Aliens	Pavel Pankov	Kyrgyzstan
	1	Flood	Luka Kalinović	Croatia
	1	Sails	Ivan Sikiric	Croatia
	2	Miners	Lovro Pužar	Croatia
	2	Pairs	Luka Kalinović	Croatia
	2	Trainings	Luka Kalinović	Croatia
2006	1	Forbidden Subgraph	Francisco Zaragoza	Mexico
	1	Pyramid	Hugo Ryckeboer	Argentina
	1	Deciphering the Mayan Writing	Cesar Cepeda	Mexico
	2	A Black Box Game	Cesar Cepeda, Hugo Ryckeboer	Mexico
	2	The Valley of Mexico	Francisco Zaragoza, Feliú Sagols, Criel Merino	Mexico
	2	Joining Points	Criel Merino	Mexico
2005	1	Garden	Mathias Hiron, Szymon Acedański	France
	1	Mean Sequence	Tom Verhoeff, Szymon Acedański	the Netherlands
	1	Mountain	Jakub Pawlewicz, Anna Niewiarowska	Poland
	2	Birthday	Jakub Pawlewicz	Poland
	2	Rectangle Game	Jakub Radoszewski, Marcin Kubica	Poland
	2	Rivers	Łukasz Kowalik	Poland
2004	1	Artemis	Bruce Merry	South Africa
	1	Hermes	Elias Koutsoupas	Greece
	1	Polygon	Yannis Emiris	Greece
	2	Empodia	Yannis Emiris	Greece
	2	Farmer	unknown	unknown
	2	Phidias	Vasilis Zisimopoulos	Greece

Year	Day	Task	Author(s)	Country
2003	1	Trail Maintenance	Local SC	unknown
	1	Comparing Code	Local SC	unknown
	1	Reverse	Tom Verhoeff	the Netherlands
	2	Guess Which Cow	Local SC	unknown
	2	Amazing Robots	unknown	unknown
	2	Seeing the Boundary	Local SC	unknown
2002	1	The Troublesome Frog	Soo-Hwan Kim, Greg Galperin	South Korea, USA
	1	Utopia Divided	Sergejs Meln, iks, Jung-Hum Park, Chong-Dae Park, Kee-Moon Song, Ian Munro	Latvia, South Korea, Canada
	1	XOR	Hwan-Gue Cho, Chong-Dae Park, Jyrki Nummenmaa	South Korea, Finland
	2	Batch Scheduling	Hee-Chul Kim, Jyrki Nummenmaa	South Korea, Finland
	2	Bus Terminals	Chan-Su Shin, Djura Paunić	South Korea, Serbia
	2	Two Rods	Hwan-Gue Cho, Ian Munro	South Korea, Canada
2001	1	Mobile Phones	Timo Tossavainen, Jyrki Nummenmaa	Finland
	1	Ioiwari Game	Gyula Horváth	Hungary
	1	Twofive	Sergejs Meln, iks, Tero Karras	Latvia, Finland
	2	Score	Timo Poranen, Jyrki Nummenmaa	Finland
	2	Double Crypt	Tom Verhoeff	the Netherlands
	2	Depot	Jyrki Nummenmaa, Erkki Mäkinen	Finland
2000	1	Palindrome	Erkki Mäkinen ⁶	Finland
	1	Car Parking	Lin Xiao	China
	1	Median Strength	Tom Verhoeff	the Netherlands
	2	Walls	Hong Wang, Donglin Xia	China
	2	Post Office	Hong Wang, Donglin Xia	China
	2	Building with Blocks	Hong Wang, Donglin Xia	China
1999	1	Little Shop of Flowers	Halit Oğuztüzün	Türkiye
	1	Hidden Codes	Faruk Polat	Türkiye
	1	Underground City	Göktürk Üçoluk	Türkiye
	2	Traffic Lights	İsmail Hakkı Toroslu	Türkiye
	2	Flatten	İsmail Hakkı Toroslu	Türkiye
	2	A Strip of Land	İsmail Hakkı Toroslu	Türkiye

Year	Day	Task	Author(s)	Country
1998	1	Contact	Luís Caires	Portugal
	1	Starry Night	Artur Miguel Dias	Portugal
	1	Party Lamps	Nuno Mamede	Portugal
	2	Picture	Manuel Próspero dos Santos	Portugal
	2	Camelot	Luís Caires	Portugal
	2	Polygon	Margarida Mamede	Portugal
1997	1	Mars Explorer	unknown	unknown
	1	Game of Hex	unknown	unknown
	1	Toxic iShongololo	unknown	unknown
	2	Map Labelling	unknown	unknown
	2	Character Recognition	unknown	unknown
	2	Stacking Containers	unknown	unknown
1996	1	A Game	Gyula Horváth	Hungary
	1	Job Processing	Gyula Horváth	Hungary
	1	Network of Schools	Gyula Horváth	Hungary
	2	Sorting a Three-Valued Sequence	Gyula Horváth	Hungary
	2	Longest Prefix	Gyula Horváth	Hungary
	2	Magic Squares	Gyula Horváth	Hungary
1995	1	Packing Rectangles	Peter Kluit	the Netherlands
	1	Shopping Offers	Jacco Gnodde	the Netherlands
	1	Printing	Cecile Crutzen	the Netherlands
	2	Letter Game	Conny Veugen	the Netherlands
	2	Street Race	Pim van den Broek	the Netherlands
	2	Wires and Switches	Tom Verhoeff	the Netherlands
1994	1	The Triangle	unknown	unknown
	1	The Castle	unknown	unknown
	1	The Primes	unknown	unknown
	2	The Clocks	unknown	unknown
	2	The Buses	unknown	unknown
	2	The Circle	unknown	unknown
1993	1	Necklace	unknown	unknown
	1	Company	unknown	unknown
	1	Rectangles	unknown	unknown
	2	Canadian Airlines	unknown	unknown

⁶Following multiple exchanges with committee members from China, Hungary, the Netherlands, and Finland, we believe that Erkki Mäkinen was probably the author; however, this is not 100% certain.

Year	Day	Task	Author(s)	Country
1992	1	Islands in the Sea	unknown	United Kingdom
	2	Climbing a Mountain	unknown	China
1991	1	Square	unknown	unknown
	2	S-Terms	unknown	unknown
1990	1	Schedule	unknown	unknown
	2	Robots	unknown	unknown
1989	1	Boxes	Wenhu Wu ⁷	China

⁷Based on available records, Wenhu Wu of China is believed to be the probable author of this task.

B Complete Catalogue of EGOI Task Authors

Table 10 presents the full EGOI catalogue. Complete authorship records exist for every task.

Table 10: Complete catalogue of EGOI task authors, 2021–2025.

Year	Day	Task	Author(s)	Country
2025	1	Gift Boxes	Anja Dožić	Serbia
	1	Dark Ride	Ivan Gaspardy	New Zealand
	1	Monster Go	Darío Martínez Ramírez, Manuel Torres Cid	Spain
	1	Wind Turbines	Chur Zhe Yaw, Shi Wei Tia	Singapore
	2	String Problem	Yoav Linhart	Israel
	2	Currents	Brian Lee Jun Siang	Singapore
	2	IMO	Eliška Macáková	Slovakia
	2	Laser Strike	Luca Versari	Italy
2024	1	Infinite Race	Nils Gustafsson	Sweden
	1	Bouquet	Jasmin Studer	Switzerland
	1	Team Coding	Yann Viegas	France
	1	Garden Decorations	Massimo Cairo	Italy
	2	Circle Passing	Hazem Issa	Egypt
	2	Bikeparking	Nils Gustafsson	Sweden
	2	Light Bulbs	Viktor Kozhuharov	Bulgaria
	2	Make them Meet	Hazem Issa	Egypt
2023	1	Inflation	Isaac Chan	Hong Kong
	1	Padel Prize Pursuit	Pavle Martinović, Mladen Puzić	Serbia
	1	Find the Box	Nils Gustafsson	Sweden
	1	Bikes vs Cars	Nils Gustafsson	Sweden
	2	Carnival General	Nils Gustafsson	Sweden
	2	Candy	Yann Viegas	France
	2	Sopsug	Jakub Tarnawski	Poland
	2	Guessing Game	Edward Xiao	Canada
2022	1	SubsetMex	Hazem Issa	Egypt
	1	Lego Wall	Michal Švagerka	Czech Republic
	1	Social Engineering	Nils Gustafsson	Sweden
	1	Tourists	Stanisław Czech	Poland
	2	Data Centers	Cheng Zhong	China
	2	Superpiece	Michal Švagerka	Czech Republic
	2	Toy Design	Petr Mitrichev	Switzerland
	2	Chika Wants to Cheat	Vlad Gavrilă	Romania

Year	Day	Task	Author(s)	Country
2021	1	Number of Zeros	Jakub Tarnawski	Poland
	1	Luna likes Love	Mladen Puzić	Serbia
	1	Twin Cookies	Ofer Wald	Israel
	1	Lanterns	Benjamin Qi, Richard Qi, Siyong Huang	USA
	2	Shopping Fever	Monika Steinová	Slovakia
	2	Railways	Antti Röyskö	Finland
	2	Angry Cows	Nils Gustafsson	Sweden
	2	Double Move	Petr Mitrichev	Switzerland

C Top Authors by Raw Task Count

Table 11 ranks authors by the number of tasks they authored or co-authored, without adjusting for co-authorship. Compare with Table 1 (on page 4) in the main text.

Table 11: Most prolific task authors by raw task count.

Rank	Author	Tasks	Country	Years	Span
1	Hirota Yoneda	7	Japan	2022–2025	4
	Masataka Yoneda	7	Japan	2022–2025	4
	Gyula Horváth	7	Hungary	1996–2001	6
4	Michal Forišek	6	Slovakia	2010–2017	8
5	Tom Verhoeff	5	the Netherlands	1995–2005	11
	Jyrki Nummenmaa	5	Finland	2001–2002	2
7	Monika Steinová	4	Slovakia	2010–2017	8
	Richard Peng	4	Canada, China, USA	2008–2013	6
	Gordon Cormack	4	Canada	2010	1
	Velin Tzanov	4	Bulgaria	2008–2009	2
11	Félix Moreno Peñarrubia	3	Spain	2022–2025	4
	Saeed Seddighin	3	Iran	2017–2019	3
	Aleksandar Ilić	3	Serbia	2008–2017	10
	Mihai Pătraşcu	3	Romania	2010–2011	2
	Carl Hultquist	3	South Africa	2009	1
	Luka Kalinović	3	Croatia	2007	1
	Hong Wang	3	China	2000	1
	Donglin Xia	3	China	2000	1

D Complete List of Female-Authored Tasks

Table 12 lists every task at either the IOI or EGOI that involved at least one female author.

Table 12: All IOI and EGOI tasks with at least one female author.

Comp.	Year	Task	Author(s)	Country
IOI	2018	Werewolf	Mohammad Roghani, Helia Ziaei	Iran
IOI	2017	Nowruz	Monika Steinová, Michal Forišek, HSC	Slovakia
IOI	2015	Boxes with Souvenirs	Monika Steinová	Slovakia
IOI	2013	Game	Monika Steinová	Slovakia
IOI	2010	Maze	Monika Steinová, Michal Forišek	Slovakia
IOI	2005	Mountain	Jakub Pawlewicz, Anna Niewiarowska	Poland
IOI	1998	Polygon	Margarida Mamede	Portugal
IOI	1995	Printing	Cecile Crutzen	the Netherlands
IOI	1995	Letter Game	Conny Veugen	the Netherlands
EGOI	2025	Gift Boxes	Anja Dožić	Serbia
EGOI	2025	IMO	Eliška Macáková	Slovakia
EGOI	2024	Bouquet	Jasmin Studer	Switzerland
EGOI	2021	Shopping Fever	Monika Steinová	Slovakia

E IOI Task Authors Who Were Former Contestants

Of the 135 known IOI task authors, 66 were themselves IOI contestants at some point. Table 13 lists these individuals, sorted alphabetically by surname. The δ column shows the number of years between their final contest appearance and their first authored task.

Table 13: IOI task authors who were former IOI contestants.

Author	Country	Contestant	Author	δ
Chethiya Abeyasinghe	Sri Lanka	2002–2007	2016	9
Maxim Akhmedov	Russia	2012	2021	9
Pikatan Arya Bramajati	Indonesia	2020, 2021	2024	3
Cesar Cepeda	Mexico	1993	2006	13
Arthur Charguéraud	France	2002	2013	11
Christopher Chen	Australia	2005–2007	2010	3
Gleb Evstropov	Russia	2010	2016	6
Jittat Fakcharoenphol	Thailand	1992	2011	19
Alireza Farhadi	Iran	2012	2019	7

Author	Country	Contestant	Author	δ
Martin Fixman	Argentina	2008–2010	2011	1
Long Fan	China	2005, 2006	2009	3
Michal Forišek	Slovakia	1998, 1999	2010	11
Daniel Graf	Switzerland	2009	2017	8
Agustín Santiago Gutiérrez	Argentina	2006, 2007	2021	14
Vytautas Gruslys	Lithuania	2006–2008	2013	5
Luke Harrison	Australia	2010	2012	2
Ta-Jui Ho	Taiwan	2016	2021	5
Kazuhiro Hosaka	Japan	2008, 2009	2013	4
Weidong Hu	China	2004, 2005	2015	10
Carl Hultquist	South Africa	1999, 2000	2009	9
Aleksandar Ilić	Serbia	2002, 2003, 2005	2008	3
Andreja Ilić	Serbia	2006	2012	6
Hazem Issa	Egypt	2019–2021	2022	1
Peyman Jabbarzade	Iran	2015	2019	4
Sunghyeon Jo	South Korea	2014, 2015	2020	5
Luka Kalinović	Croatia	2002–2004	2007	3
Adam Karczmarz	Poland	2009	2015	6
Tero Karras	Finland	1997–2000	2001	1
Christian Kauth	Luxembourg	2003, 2004	2011	7
Riku Kawasaki	Japan	2016, 2017	2018	1
Alireza Keshavarz	Iran	2021	2023	2
Eryk Kopczyński	Poland	1998	2015	17
Richard Kráľovič	Slovakia	1997–1999	2012	13
Jakub Łącki	Poland	2004, 2005	2014	9
Joshua Lau	Australia	2012, 2013	2024	11
Xiao Mao	China	2017	2020	3
Félix Moreno Peñarrubia	Spain	2018	2022	4
Pavle Martinović	Serbia	2018	2021	3
Bruce Merry	South Africa	1996–2001	2004	3
Jonathan Mosheiff	Israel	2003–2005	2014	9
Shogo Murai	Japan	2010–2012	2018	6
Arthur Nascimento	Brazil	2014	2024	10
Danylo Mysak	Ukraine	2005	2019	14
Jakub Pawlewicz	Poland	1994–1996	2005	9
Mihai Pătrașcu	Romania	1999–2001	2010	9
Richard Peng	Canada, USA, China	2004–2006	2008	2
Amaury Pouly	France	2005	2013	8

Author	Country	Contestant	Author	δ
Kevin Luiz Ponte Pucci	Portugal	2017–2019	2022	3
Lovro Pužar	Croatia	2002, 2004	2007	3
Mikhail Pyaderkin	Russia	2010	2018	8
Angus Ritossa	Australia	2017–2019	2020	1
Ammar Fathin Sabili	Indonesia	2013	2018	5
Saeed Seddighin	Iran	2007	2017	10
Tadija Šebez	Serbia	2018–2020	2021	1
Ranald Lam Yun Shao	Singapore	2012–2014	2020	6
Bernard Teo	Singapore	2012, 2013	2023	10
Ivan Sikiric	Croatia	2001, 2002	2007	5
Kee-Moon Song	South Korea	1999	2002	3
Mohamed Taha	Egypt	2001	2008	7
Velin Tzanov	Bulgaria	2001, 2002	2008	6
Hamed Valizadeh	Iran	2012, 2013	2017	4
Normunds Vilciņš	Latvia	2009, 2010	2011	1
Masaki Watanabe	Japan	2006	2008	2
Hiroataka Yoneda	Japan	2018, 2020	2022	2
Masataka Yoneda	Japan	2018–2020	2022	2
Lim Rui Yuan	Singapore	2020, 2021	2022	1

F EGOI Task Authors Who Were Former Contestants

Seventeen EGOI task authors were former contestants at either the IOI or the EGOI (or both). Table 14 (below) lists these individuals, sorted alphabetically by surname. IOI contestant years are marked with *.

Table 14: EGOI task authors who were former contestants. IOI contestant years are marked with *.

Author	Country	Contestant	Author	δ
Massimo Cairo	Italy	2007–2010*	2024	14
Anja Dožić	Serbia	2021–2024	2025	1
Siyong Huang	USA	2021*	2021	0
Hazem Issa	Egypt	2019–2021*	2022, 2024	1
Viktor Kozhuharov	Bulgaria	2019*, 2021*	2024	3
Eliška Macáková	Slovakia	2021–2024*, 2021–2024	2025	1
Pavle Martinović	Serbia	2018*	2023	5
Petr Mitrichev	Switzerland	2000–2002*	2021, 2022	19

Author	Country	Contestant	Author	δ
Mladen Puzić	Serbia	2019*, 2020*	2021, 2023	1
Benjamin Qi	USA	2018*, 2019*	2021	2
Richard Qi	USA	2021*	2021	0
Antti Röyskö	Finland	2017*	2021	4
Jasmin Studer	Switzerland	2023*, 2021–2023	2024	1
Manuel Torres Cid	Spain	2022*	2025	3
Luca Versari	Italy	2012*	2025	13
Yann Viegas	France	2022*	2023, 2024	1
Edward Xiao	Canada	2022*	2023	1

G Host Countries

Table 15 lists the host country for each IOI and EGOI edition.

Table 15: Host countries for the IOI (1989–2025) and EGOI (2021–2025).

IOI		IOI (cont.)		EGOI	
Year	Host	Year	Host	Year	Host
1989	Bulgaria	2008	Egypt	2021	Switzerland
1990	Soviet Union	2009	Bulgaria	2022	Türkiye
1991	Greece	2010	Canada	2023	Sweden
1992	Germany	2011	Thailand	2024	the Netherlands
1993	Argentina	2012	Italy	2025	Germany
1994	Sweden	2013	Australia		
1995	the Netherlands	2014	Taiwan		
1996	Hungary	2015	Kazakhstan		
1997	South Africa	2016	Russia		
1998	Portugal	2017	Iran		
1999	Türkiye	2018	Japan		
2000	China	2019	Azerbaijan		
2001	Finland	2020	Singapore		
2002	South Korea	2021	Singapore		
2003	USA	2022	Indonesia		
2004	Greece	2023	Hungary		
2005	Poland	2024	Egypt		
2006	Mexico	2025	Bolivia		
2007	Croatia				

References

- Azalov, P. (1989). Bulgarian competitions on informatics (computing programming). In Hoekman, H. and Pokrovsky, A., editors, UNESCO International Science Olympiads, pages 58–63.
- Burton, B. A. & Hiron, M. (2008). Creating Informatics Olympiad tasks: exploring the black art. *Olympiads in Informatics*, 2:16–36.
- Forišek, M. (2013). Pushing the boundary of programming contests. *Olympiads in Informatics*, 7:23–35.
- Kalinicenko, E. & Opmanis, M. (2016). Collecting, processing and maintaining IOI statistics. *Olympiads in Informatics*, 10:161–176.
- Kenderov, P. S. (2017). Three decades of international informatics competitions (how did IOI start?). *Olympiads in Informatics*, 11 (Special Issue):3–10.
- Maggiolo, S. (2015). An update on the female presence at the IOI. *Olympiads in Informatics*, 9:127–137.
- Marrone Berzetti di Buronzo, L. & Gambirasio, N. (2025). Girls in STEM: a qualitative analysis of factors and actors impacting on girls' engagement in international computer science competitions. *Olympiads in Informatics*, 19:101–114.
- Schrijvers, E. (2024). IOI task explanation videos [YouTube channel]. <https://youtu.be/PwC2o1-Hjf08>
- Steegh, A. M., Hoffmann, N., Hohn, M., & Parchmann, I. (2019). Gender differences in mathematics and science competitions: a systematic review. *Journal of Research in Science Teaching*, 56(10):1431–1460.
- Vasiga, T., Cormack, G., & Kemkes, G. (2008). What do olympiad tasks measure? *Olympiads in Informatics*, 2:181–191.
- Verhoeff, T. (2009). 20 years of IOI competition tasks. *Olympiads in Informatics*, 3:149–166.
- Yamaguchi, R. S. & Ito, T. (2024). The first step towards increasing female participants in the Olympiads in Informatics in Japan. *Olympiads in Informatics*, 18:185–193.
- Yang, Y., Tian, T. Y., Woodruff, T. K., Jones, B. F., & Uzzi, B. (2022). Gender-diverse teams produce more novel and higher-impact scientific ideas. *Proceedings of the National Academy of Sciences*, 119(36):e2200841119.

About the Author

Eljakim Schrijvers, though most people just call him Kim (the “elja” is silent, apparently), is deeply involved in the IOI community. He has been part of every EGOI since its start in 2021 and was the organiser of EGOI 2024 in the Netherlands.

He also contributes behind the scenes, having served for many years as treasurer of the IOI. On the technical side, he runs contest systems for multiple Bebras and Kangourou Sans Frontières (KSF) countries, and has built the registration database used by many contests.

Kim also runs a YouTube channel where he explains algorithms in short, accessible videos, sometimes with more enthusiasm (and content) than production quality.

When he's not working on his company, organising contests, or talking about algorithms, he can occasionally be found on a golf course, though not very successfully, by his own admission. He insists he enjoys it anyway.

On the Recurrent Themes of Constructives and Interactives in Informatics Olympiads

Troy Dylan SERAPIO¹, EdRonn James PANTI², Farrell Eldrian WU³

¹Philippine Science High School Main Campus, Quezon City, National Capital Region, Philippines

²Philippine Science High School Main Campus, Quezon City, National Capital Region, Philippines

³Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America

e-mail: b2027tdserapio@pshs.edu.ph, b2026ejtpanti@pshs.edu.ph, farrellw@mit.edu

Abstract. Constructive and interactive tasks appear regularly in major informatics olympiads. Yet, they are often treated as ad hoc extensions of standard paradigms rather than as domains with recurring reasoning structures. This paper studies a curated set of constructive and interactive problems drawn from TOKI (Tim Olimpiade Komputer Indonesia), SGNIO (Singapore National Olympiad of Informatics), and EGOI (European Girl’s Olympiad in Informatics) archives. For each selected task, we prepared an abridged solution write-up and extracted the underlying theme (central conceptual mechanism) and heuristic (meta-level reasoning pattern that guides discovery). Synthesizing across the curated set yields three recurring themes (i.e., Balanced Queries, Decomposition of Graphs, and Iterative Candidate Refinement) and five problem-solving heuristics. To concretize such themes, we present representative problems illustrating them and discuss how these themes characterize constructives and interactives beyond surface-level algorithms and data structures.

Keywords: competitive programming; olympiads; interactive problems; constructive problems; thematic analysis; problem-solving heuristics

1. Introduction and Background

1.1 *Background on Constructive and Interactive Problems in Informatics Olympiads*

Competitive programming has long served as a rigorous domain for algorithmic reasoning and computational problem solving. Over time, the diversity of problems encountered in contests has led to the formalization of well-known categories such as dynamic programming, graph theory, and greedy algorithms — each extensively analyzed and systematized in both educational and research contexts. Yet, not all kinds of problems have received the same degree of theoretical attention. Among these are constructive and interactive problems, categories that appear regularly in contests but remain largely dismissed as extensions of existing paradigms.

A constructive problem is focused on the explicit construction of an object, structure, or sequence that satisfies a given set of constraints. This includes designing a valid graph, permutation, or configuration. To be classified as a “constructive,” the main challenge of the task lies in synthesizing a structure that fulfills the required conditions. In contrast, an interactive problem involves dynamic information exchange between the contestant’s program and the judge, often requiring the program to adapt as new data is revealed by the judge.

However, despite their prevalence in major olympiads such as the IOI and ICPC, both remain underexplored, especially as unified domains of study. Thus, this paper examines curated problems from TOKI, SGNOI, and EGOI to identify recurring themes and heuristics that concretize the reasoning frameworks of these two conceptually-rich yet underexplored classes of problems.

1.2 Research Questions

This paper presents an exploratory qualitative analysis of constructive and interactive problems in competitive programming, particularly identifying recurring themes and problem-solving heuristics that characterize this class of problems. In this context, a theme refers to the central conceptual structure underlying a problem’s solution, while a heuristic denotes the meta-strategies that guide the solver toward that idea.

For the identification of themes, we first create concise solution write-ups for each problem to ensure full understanding of the solutions. We then apply inductive thematic analysis (Braun & Clark, 2022) to these write-ups together with the official editorials of representative problems curated from TOKI, SGNOI, and EGOI archives. The objective of this analysis is to systematically examine how solutions are structured, identify recurring conceptual patterns, and synthesize these patterns into a small set of overarching themes that capture the nature of constructive and interactive problems.

For the identification of heuristics, we use heuristic inquiry (Moustakas, 1990) grounded in the lived experiences of the authors as competitive programmers. This process involves reflective examination of the authors’ problem-solving experiences in the curated problem archive to propose practical heuristics that align with the identified themes. The goal is not to claim universal strategies, but to exhibit meta-strategies that may guide solvers when approaching constructive and interactive tasks.

1.3 Data Processing Procedures

With the objectives outlined above, we examined problem archives from the following sources: TOKI (Tim Olimpiade Komputer Indonesia), SGNOI (Singapore National Olympiad in Informatics) from 2017 to the present, and EGOI (European Girls’ Olympiad in Informatics). Problems were initially identified through contest archives and platform tags indicating constructive or interactive problem types. TOKI problems were filtered directly

using the constructive and interactive tags on the TLX platform, while SGNOI and EGOI problems were manually screened from their respective archives on `codebreaker.xyz` and `QOJ.ac`.

From this initial pool, we applied a second set of criteria: interactive problems must involve adaptive interaction with a judge; constructive problems must require producing a valid configuration where the main difficulty lies in devising the construction rather than optimizing bounds; problems exhibiting fundamentally different or heterogeneous solution paradigms were excluded to maintain consistency in analysis; problems had to be non-trivial, typically containing multiple subtasks and requiring more than a single isolated trick; and problems exceeding an estimated difficulty of around 2000 on the Codeforces scale were excluded unless accompanied by a clear editorial. This process resulted in a final curated set of sixteen TOKI problems, three SGNOI problems, and three EGOI problems.

For every selected problem, we prepared a concise write-up summarizing the core solution and reducing it to its essential conceptual structure. These write-ups were primarily developed by the lead author, with additional contributions from co-authors on selected problems, and served as the primary data for our inductive thematic analysis. Instead of applying predefined categories, we first distilled each solution into its conceptual “barebones,” grouped solutions that shared similar reasoning structures, and iteratively refined these groupings into three overarching themes that capture recurring patterns in constructive and interactive problem solving. Additional problems, beyond what will be thoroughly discussed in this paper, may be found in Appendix B.

In parallel, we conducted a heuristic inquiry grounded in our lived experiences as competitive programmers. Through reflective examination of the reasoning processes involved in solving and analyzing the curated problems, we synthesized five heuristics aligned with the identified themes. Each pairing of Theme and Heuristic was supported by a brief analytical justification explaining its relevance. Our goal is to emphasize conceptual reasoning and solver decision-making rather than implementation details, guiding the reader through the structure and motivation behind constructive and interactive solutions while maintaining an analytical and accessible presentation style. Some common terminologies in competitive programming are used for brevity and familiarity; concise definitions are provided in Appendix A.

2. Balanced Queries

In many interactive problems, the judge’s responses are drawn from a small, discrete set, with binary feedback such as {yes, no} being a common example. The central idea behind balanced querying is to design each query so that all possible responses yield comparable amounts of information about the hidden state. By ensuring that no outcome is significantly more informative than another, each interaction contributes meaningfully toward narrowing the search space, reducing the worst-case total number of queries needed to reach a solution. Problems employing this theme often rely on reasoning analogous to

binary search or information-theoretic decision making, where the objective is to partition the remaining candidate space into subsets of roughly equal size in order to maximize expected information gain.

2.1 *Maybe Guess the Number (OSN Informatika 2015, Day 1C)*

The first problem that we will tackle is called “Maybe Guess The Number” from the TOKI archive. To summarize this problem, Kwek must determine an unknown integer x between 1 and n by asking at most k questions. In each question, Kwek submits a set of integers S , where S is a subset of the numbers from 1 to n , and Kwak responds with one of three possible answers: YES if x is in S , MAYBE if x is not in S but either $x - 1$ or $x + 1$ is in S , and NO otherwise. After at most k queries, Kwek must correctly identify the value of x . It is implicitly stated that a solution that runs with query complexity $O(k) \leq O(\log N)$ is required to obtain full points.

The key observation lies in analyzing the behavior of an arbitrary queried set S , shown in the list below:

- If $x \in S$, the response will be YES. This indicates that one of the elements inside the set is definitely x . Thus, we can call all of the elements “YES numbers” (as they collectively triggered the YES response).
- If $x \pm 1 \in S$, the response will be MAYBE. This means that one of the elements adjacent to those in the set (but not in the set itself) is definitely x . Thus, we can call all of the elements “MAYBE numbers.”
- Otherwise, the response will be NO, signifying that none of the elements in or adjacent to S can be x . Thus, we can call all of the elements “NO numbers.”

Once these notions of labels on numbers are established, the guessing game transforms into partitioning a search space (particularly, the number line). The central idea is to design each query so that the three possible labels of numbers (YES, MAYBE, NO) partition the numbers as evenly as possible. Thus, a possible balanced pattern (on some number line) is shown below:

[ABBA][ABAB ... AB][CC ... CC]

where A means “YES,” B means “MAYBE,” and C means “NO.” The set queried to the judge is composed only of numbers with labels A , while the other labels follow as consequences of adjacency. After each query, all numbers inconsistent with the judge’s response are eliminated, and the remaining candidates are recolored according to the same scheme. Since every round reduces the search space by approximately a constant factor, the total number of queries grows logarithmically with respect to n , yielding a query complexity of $O(\log_3 n) \sim O(\log n)$.

This reasoning illustrates the Balanced Queries theme: each query is designed to maximize guaranteed information by minimizing the disparity among possible outcomes (YES, MAYBE, NO). Specifically, the search space is partitioned into three equally sized subsets, each corresponding to one response. As a result, regardless of the outcome, each interac-

tion reduces the remaining candidates by roughly the same proportion, ensuring that the querying process remains balanced.

2.2 One-Day Delivery (OSN Informatika 2023, Day 1C)

An unknown shoe size s , where $l \leq s \leq n$ must be determined by the night of the day t . Each morning, you may make exactly one shoe of an integer size at most n and ship it; the shoe arrives on the night of the following day. Upon arrival, you receive a message corresponding to the shipment from the previous day: KEKECILAN if the shoe is too small, KEBESARAN if the shoe is too big, or PAS if the shoe fits, in which case the program immediately ends with an accepted verdict. The task is to design a daily schedule of shoe sizes that guarantees the correct size will be identified by night t , despite the one-day delay in feedback due to the query complexity $O(\log N)$.

We need to identify a hidden integer $x \in [l, n]$ using ternary feedback: KEKECILAN (our guess is too small), KEBESARAN (too large), or PAS (correct). The twist is that feedback is delayed by one turn: after issuing a query q_t , we only receive the judge's response to $q_{(t-1)}$. This means we cannot immediately update the search interval based on the number we just asked, thus making a standard binary search break (this only gives 16/100 points in the original task).

To cope with the one-step delay, we structure each “decision” through two consecutive queries that effectively partition the current search space $[l, r]$ into three blocks:

$$[l, a-1] \quad a \quad [a+1, b-1] \quad b \quad [b+1, r]$$

where we ask a first and b next. Once the delayed response arrives, it tells us whether the secret is to the left, between, or to the right of these pivots (or equals one of them). The key is to choose a (and hence how the interval is split) so that no matter which delayed outcome occurs, the remaining uncertainty is as small as possible given that we will already have spent (or be spending) two queries.

The optimal way to place these pivots follows a Fibonacci-style partition: maintain interval lengths that decrease according to consecutive Fibonacci numbers. Intuitively, because each refinement consumes one new query while “processing” the previous response, the worst-case remaining interval after a delayed step should match the recurrence of the Fibonacci search. Implement this via a helper $getNext(l, r)$ that returns the next query position using the largest Fibonacci decomposition fitting $[l, r]$, ensuring the three resulting regions are sized so that the worst-case branch remains optimal.

Algorithmically, keep the current feasible range $[l, r]$ and a variable holding the previous response. Repeatedly compute the next query $q = getNext(l, r)$ and print q ; then read the response to the previous query and shrink $[l, r]$ accordingly: if it is KEBESARAN, set $r = q_{prev} - 1$; if KEKECILAN, set $l = q_{prev} + 1$; if PAS, stop. The Fibonacci placement guarantees that, despite the one-turn lag, each new query is “pre-scheduled” so that every possible delayed outcome leaves an optimally small remaining search interval.

3. Decomposition of Graphs

Many constructive problems operate by decomposing a complex graph or structure into simpler subcomponents, solving each subcomponent individually, and then reassembling them into a complete solution. This approach often starts with identifying simpler fragments that can be independently handled. Once these components satisfy local constraints, they are combined to form a valid global construction. This theme captures the reasoning process of reducing a global construction problem into a family of simpler local constructions whose properties collectively imply the general case.

3.1 Autosynthesis (COMPFEST 15 Final CPC Junior 2A)

Chaneka is given an array A of n positive integers, initially with no elements circled. In each operation, she may circle any element of A , possibly circling the same index multiple times. After all operations, she constructs a sequence R consisting of all uncircled elements of A in increasing order of indices, and a sequence P where P_i denotes the index circled in the i -th operation. Determine whether it is possible to choose operations so that $R=P$, or report that it is impossible; if multiple solutions exist, output any one. From the problem statement, the required algorithmic time complexity to earn full points is $O(N)$.

We model the array as a functional graph with nodes $1, \dots, n$, where each node i has a directed edge i to $A[i]$. For an edge u to v , we call u a parent of v and v a child of u . Thus, each node will be labeled either CIRCLED (belongs in P) or UNCIRCLED (belongs in R). This leads to the following set of rules by translating the array representation into the functional graph representation:

- If all parents of a node k are circled, then all array elements with value k are removed from R , so k they cannot appear in P ; hence, k must be uncircled.
- If at least one parent of k is uncircled, then k appears in R , so it must also appear in P , implying that k must be circled.

These rules allow us to determine the coloring for all nodes reachable from any node with an in-degree 0. We start from such nodes and perform a BFS, propagating colors using the rules above. Any remaining unvisited nodes must lie in connected components that are purely cycles. In these components, nodes must be colored alternately CIRCLED and UNCIRCLED; this is possible if and only if the cycle length is even, while odd cycles make the condition $R=P$ impossible. The entire procedure runs in $O(N)$ time (BFS and iterating through the cycles).

This task follows the idea of graph decomposition by breaking the global requirement $R=P$ into simpler, locally checkable conditions on the functional graph defined by i to $A[i]$. The graph naturally decomposes into components reachable from nodes with in-degree 0 and components that are purely cycles. The former can be handled independently using a BFS that propagates forced colorings based only on parent-child relationships. At the same time, the latter reduces to standalone cycle problems whose validity depends solely on parity. Each component is solved in isolation using local rules, and the full construction is

obtained by combining these solutions, making the overall approach a direct application of decomposing a global construction into simpler subgraph-level constructions.

3.2 Social Engineering (EGOI 2022 Day 1(3))

A social network is modeled as a connected undirected graph with n vertices and m edges, where each edge may be used at most once. Starting from Maria at vertex 1, players alternately challenge a friend by traversing an unused edge, forming a walk that never repeats an edge. A player loses if they have no legal move on their turn. Maria always moves first and plays optimally whenever she has a winning strategy, while the remaining players cooperate to force her to lose if possible. Given the full graph in advance, the task is to determine whether Maria has a forced win, or otherwise to coordinate the friends' moves so that she is eventually left without a legal move. The algorithm must correctly handle all optimal Maria strategies and run in $O(n+m)$.

To solve this problem, the first step is to model the game directly as the given graph, with Maria fixed at vertex 1. The neighbors of Maria are the only vertices through which play can move between her position and the rest of the graph, so we call them entry vertices. Removing Maria partitions the graph into connected components, each containing some number of entry vertices. The key observation is that the outcome depends solely on the parity of these entry vertices within each component. If any component contains an odd number of entry vertices, then Maria has a winning strategy. Once she enters such a component, parity guarantees that the remaining players will eventually run out of moves first. In this case, no coordinated response can prevent her from winning.

Conversely, if every component contains an even number of entry vertices, then the remaining players can force Maria to lose. The strategy is to pair the entry vertices within each component so that each pair is connected by a path that does not conflict with other pairs. Whenever Maria enters a component through some entry vertex, the friends respond by following the precomputed path to its paired entry vertex and then returning the move to Maria. This enforces a controlled entry-and-exit structure within each component and prevents Maria from creating a position where the friends are trapped without a reply.

To construct these pairings efficiently, it suffices to work on a spanning tree of each component. For every component, select an arbitrary spanning tree and perform a depth-first search. Each subtree pairs its entry vertices locally and propagates at most one unmatched entry vertex to its parent; subtrees containing an even number of entry vertices propagate none. This process succeeds if and only if the total number of entry vertices in the component is even. The pairing construction and the resulting strategy can be implemented within a single DFS traversal per component, giving an overall time complexity of $O(n+m)$.

This solution illustrates graph decomposition as a unifying principle. Removing Maria separates the global game into independent components whose outcomes are determined solely by parity conditions. Each component is then reduced to a spanning tree, since additional edges do not affect the pairing argument. By solving each component independently

and combining the results, the complex interactive game reduces to a collection of simple structural checks and local constructions, leading to an elegant strategy for defeating Maria.

4. Iterative Candidate Refinement

Many constructive and interactive problems begin by proposing one or more candidate entities that could satisfy the constraints or goal of the problem. The solver then iteratively tests, eliminates, or refines these candidates until a correct or optimal one emerges. This theme often appears in problems where the search space is too large for full brute-force enumeration, but structured enough to allow progressive elimination. This idea closely parallels algorithmic search in artificial intelligence, where each piece of feedback guides systematic reasoning and adaptive decision-making toward the solution.

4.1 Colouring Balls (CKSN Informatika 2021 Day 1C)

You are given a sequence of balls, each having an unknown color represented by an integer. Your task is to construct an array *ans* such that balls with the same color are assigned the same material, while balls with different colors are assigned different materials. You may interact with the judge using a function $query(l,r)$ that allows you to determine whether all balls in the interval $[l,r]$ have the same color. The goal is to construct a valid *ans* while achieving a query complexity of $O(N \log M)$ and a time complexity of $O(N^2)$.

Thus, the solution is as follows. We first process the balls from left to right, assigning materials incrementally. Initially, we may arbitrarily assign $ans[1]=1$, since there is no prior information. Suppose we have already correctly assigned materials for positions 1 through k , and now want to determine the material of the ball $k+1$. The key observation is the following: If there exists an index $i \leq k$ such that balls at positions i and $k+1$ share the same color, then: $query(i,k)=query(i,k+1)$. Otherwise, if the color changes at the position $k+1$, the query result must differ.

To incorporate such an insight, we maintain an auxiliary array *lki*, where $lki[m]$ stores the last index that was assigned to material m . For each existing material m , we test whether $query(lki[m],k)=query(lki[m],k+1)$. If the equality holds, then the ball $k+1$ must share the same color as the ball at $lki[m]$, and we can assign $ans[k+1]=m$. Otherwise, if no such material satisfies this condition, then the ball $k+1$ represents a new color, and we assign it a new material. To optimize the identification of the correct material, we can binary-search over the materials using the monotonicity of query outcomes as we traverse the last-known indices in increasing order. This allows us to narrow down the matching material efficiently instead of checking all previous materials.

This problem exemplifies iterative candidate refinement: for each new ball, we start with the set of all previously seen colors as potential candidates and systematically eliminate the materials that are inconsistent with query results. The algorithm never attempts to guess a color outright, but it progressively refines the candidate set using information

derived from earlier assignments. Once all invalid candidates are ruled out, the remaining one (if any) must be correct.

4.2 Detecting Gold (OSN Informatika 2019 Day 2B)

You are given a grid and an unknown pair of points containing gold. You may issue queries of the form: choose a point (x,y) , and receive the Manhattan (taxicab) distance to the nearest gold point. Using 5 queries, you must determine the exact locations of the gold points.

A key geometric observation is that the locus of points at taxicab distance T from a fixed center forms a square rotated by 45° . If the center lies on a corner of the grid, then three sides of this square lie outside the grid, and the remaining feasible points form a straight line. We exploit this by querying the four corners of the grid. Each query produces a distance, which defines a line of candidate points that must contain at least one gold point.

After the four corner queries, we obtain four such lines. By the Pigeonhole Principle, each gold point must lie at the intersection of two of these lines. Moreover, valid gold positions must correspond to opposite intersections. We can then issue a fifth query at one of the intersection points to determine which opposite pair is valid. Simple boundary checks immediately discard any intersections that fall outside the grid. Thus, the time complexity and query complexity are both $O(1)$.

This solution demonstrates iterative refinement at a geometric level. Each query dramatically reduces the space of possible gold locations: from the entire grid, to lines, to intersection points. Rather than searching exhaustively, the algorithm repeatedly restricts the candidate set using the idea of the “diagonals”, until only four possibilities remain. The final query merely disambiguates among these candidates.

5. Heuristics

In addition to the three main themes that were derived from the thematic analysis, an additional five (5) heuristics were also derived from the heuristic inquiry done on the authors. These heuristics, similar to the themes, are also given names to encapsulate the purpose of such a heuristic. Thus, these heuristics include: (1) Fake Array, (2) Reduction Heuristics, (3) Wishful Thinking, (4) Subtask Decomposition, and (5) Brute Force Heuristics. Problems that best showcase these heuristics may be found in Appendix B.

5.1 Fake Array

Some problems appear to revolve around direct array manipulation, such as updating indices, simulating transitions, or repeatedly applying rules to positions, but the array is often only an encoding of a deeper structure. The Fake Array heuristic consists of deliberately reinterpreting the array as something else, most commonly a graph. A typical case is a functional graph (for example, in Autosynthesis), where each position maps to another

position according to a given rule. Recasting the problem this way often reveals underlying structure (e.g., cycles, trees rooted at cycles, or other recognizable graph forms) which then allows the task to be solved using standard graph techniques (e.g., cycle finding, DFS, and related algorithms).

5.2 *Reduction Heuristics*

If a problem features an odd query format or an unconventional output definition, this is often a hint that the query can be reformulated into something simpler. This heuristic emphasizes rewriting the queries into a representation that is easier to analyze, sometimes by adopting an entirely different viewpoint (for instance, in *Maybe Guess The Number*, the YES, MAYBE, and NO responses can be reinterpreted as a coloring problem). Once the superficial complexity is removed, many problems become manageable when the query is understood in terms of the actual quantity it computes. This pattern appears repeatedly across solved problems and reflects the broader Computer Science principle of reduction: in long-form contests, identifying the right reduction is often an essential part of the problem itself.

5.3 *Wishful Thinking*

Rather than immediately attempting to solve the entire problem, it can be highly effective to experiment with special cases, constraints, or imagined scenarios. By assuming certain properties hold and examining their consequences, one can uncover contradictions or extract useful insights. This intentional form of “wishful thinking” frequently leads to the discovery of invariants, monotonic behavior, or tight bounds that are not obvious at first glance. It is a broadly applicable technique in competitive programming and is especially powerful in constructive and interactive problems, where creative reasoning is often required.

5.4 *Subtask Decomposition*

Decomposing a problem using its subtasks is a common competitive programming strategy, but it should be applied with discretion. In some OI-style problems, subtasks may be designed around a particular intended approach that does not match how you naturally think about the problem. In these situations, overemphasizing subtasks can obscure clearer lines of reasoning. Nevertheless, subtasks remain valuable, as they frequently highlight constraints, corner cases, or key structural observations. This is particularly evident in problems like *Social Engineering*, where early subtasks clarify what constitutes a losing configuration, and later subtasks build upon this understanding to derive a guaranteed winning strategy.

5.5 *Brute Force Heuristics*

At times, the most effective initial approach is the simplest one. Writing a brute-force solution helps illuminate how the problem behaves, reveals recurring patterns, and provides a reference point for verifying more efficient methods. Even if the brute-force approach exceeds time or query limits, it can be incrementally improved, and having a solid grasp of the naive solution is often crucial during a contest.

6. Conclusion and Further Work

This paper presents an exploratory qualitative analysis of constructive and interactive problems, with a particular focus on identifying recurring themes and solver heuristics. Overall, the study provides a structured perspective on patterns that are often recognized informally by experienced competitive programmers, while offering a clearer conceptual framework for understanding these classes of problems.

6.1 *Summary of Findings*

From the thematic analysis of curated problems, the primary observation is that many constructive and interactive tasks share a small number of recurring conceptual structures despite appearing diverse at the surface level. By distilling solution write-ups into their essential reasoning patterns, we identified three overarching themes that capture common modes of thinking across problems from TOKI, SGNOI, and EGOI. These themes highlight how problems frequently revolve around reframing queries, constructing valid configurations under implicit constraints, or exploiting structural guarantees provided by the problem statement.

Meanwhile, the heuristic inquiry grounded in our lived experiences as competitive programmers revealed five recurring heuristics that guide solver decision-making. Rather than focusing on specific algorithmic techniques, these heuristics describe higher-level reasoning patterns that help reduce complexity and expose the underlying structure of a problem. Across the analyzed problems, these strategies consistently appeared during both the initial exploration phase and the refinement of solutions, suggesting that they represent transferable modes of thought rather than isolated tricks.

6.2 *Recommendations for Further Research*

There are several directions in which this work can be extended to deepen the understanding of constructive and interactive problem-solving. As a first step, it would be valuable to expand the pool of reference problems by incorporating tasks from additional informatics olympiads, such as the IOI, as well as ICPC contests. These sources are known for their depth and diversity, and a broader dataset would help validate the generality of the proposed themes while potentially revealing new recurring patterns.

Another promising direction is to refine and extend the current thematic framework. While the three themes identified in this study already capture meaningful structural insights, further refinement or the introduction of additional categories may allow for a more precise characterization of different problem types. Any expansion, however, should preserve the generality of the themes so that they remain applicable across a wide range of contests rather than becoming overly specialized.

Finally, there is significant potential in developing a systematic classification system based on the identified themes and heuristics. A searchable index or database could be constructed to organize problems according to their underlying reasoning patterns, functioning similarly to existing problem tags but operating at a more conceptual level. Such a resource could support both learning and research by helping programmers identify problems that share similar modes of thinking.

Overall, we are satisfied with the outcomes of this work, as the process of conducting the analysis has already deepened our own understanding of constructive and interactive problems. We hope that this paper will likewise provide competitive programmers with a new perspective on these problem classes and encourage further exploration of their underlying structure.

References

- Braun, V., & Clarke, V. (2022). *Thematic analysis: A practical guide*. SAGE Publications. European Girls' Olympiad in Informatics. (n.d.). Official website. <https://egoι.org>
- Moustakas, C. (1990). *Heuristic research: Design, methodology, and applications*. SAGE Publications.
- Singapore National Olympiad in Informatics. (n.d.). SGNOI archive [Computer software]. GitHub. https://github.com/noisg/sg_noi_archive
- TLX TOKI Platform. (n.d.). TLX – Competitive Programming Training Gate. <https://tlx.toki.id>

Appendix A

For the purposes of this paper, the following terms are defined as follows:

- **Program:** The solution submitted by the participant to the online judge for evaluation.
- **Judge:** The automated system responsible for evaluating submitted programs by executing them on a predefined set of test cases and verifying the correctness of their outputs.
- **Test Cases:** Sets of input data provided to a program to produce corresponding outputs, which are then compared against the official correct outputs for validation.
- **Query:** An interaction in which the program requests additional information from the judge during the execution of an interactive problem.
- **Query Complexity:** The asymptotic behaviour (usually upper-bound) of the number of queries a program uses to solve the problem. This is expressed as a function of the input size, denoted through asymptotic notation (colloquially known as Big-O notation).
- **Time Complexity:** The asymptotic behaviour (usually upper-bound) of the number of operations a program uses to solve the problem. This is expressed as a function of the input size, denoted through asymptotic notation (colloquially known as Big-O notation).

Appendix B

This is a list of more problems that fall under the themes and heuristics discussed in this paper.

Balanced Queries:

- Toxic Gene from SGN01 2023 Finals (Task 5) | *Hint: consider a binary representation*
- Detecting the Naughty Village Thief from COMPFEST 13 Penyisihan CPC Junior | *Hint: think in 1D first*

Decomposition of Graphs:

- Sharing Lapis Talas from OSN Informatika 2023 | *Hint: solve the line graph, then what can you say about a tree?*
- Game from IOI 2014 Day 1-3 | *Hint: Solve the problem in a direct way, then consider the edge case of two components that are disconnected and try to work around that.*

Iterative Candidate Refinement:

- Mining Gold from TOKI Open Contest 2018 Day 1-1 | *Hint: what does isIntegerDistance mean in a number theory context?*
- Building Tunnels from OSN Informatika 2022 Day 1-B | *Hint: brute-force all of the possible A edges that you can find*

Fake Array:

- Lost Array from SGN01 NOI 2019 Prelims Task 2 | *Hint: do floodfill to assign a C_i for each element in the array*
- Magic from TOKI Open 2017 Day 2-1 | *Hint: there is “matching” with the deck*

Reduction Heuristics:

- Message from IOI 2024 Day 1-2 | *Hint: We need to deduce which bits aren't evil. Can we model this as a graph instead by looking at the first few sent bits of each cell?*
- Weird Chickens from OSN Informatika 2015 Day 2-C | *Hint: find the blocks of DNA first before finding the sequence of said blocks*

Wishful Thinking:

- Carnival General from EGOI 2023 Day 2-A | *Hint: consider induction from 2 generals to N generals*
- Square or Rectangle? from SGN01 2015 Prelims Task 4 | *Hint: which cells must be filled for the shape to be a square?*

Subtask Decomposition:

- Building Bridges from OSN Informatika 2022 Day 2-A | *Hint: Ignore the bridges and first break the graph into connected components with limited “connection slots”*
- Split the Attractions from IOI 2019 Day 1 | *Hint: Think about what a single well-chosen DFS cut tells you about the sizes of the resulting components.*

Brute Force Heuristics:

- Scales from IOI 2015 Day 1-2 | *Hint: $6! < 3^6$.*
- Lands and Glaciers from OSN Informatika 2017 | *Hint: what's an easy way to construct F*

Acknowledgements

We are deeply grateful for the help of Mateo Campos and Jaime Maniago for their invaluable help in curating, solving, and preparing the write-ups for the selected problems. We also extend our sincere appreciation to the entire NOI.PH community for their meaningful insights on constructive and interactive problems.

Author Information

Troy Serapio - is a Grade 11 student at the Philippine Science High School – Main Campus. He is an active participant in the National Olympiad of Informatics Philippines, placing Top 6 in the 2025 IOI Philippine Team Selection and serving as a Teaching Assistant for the NOI Year-Long Training Program (NOI-YTP). He is also deeply interested in artificial intelligence and founded Tomorrow, a student-led initiative dedicated to promoting accessible and responsible AI education.



EdRonn James Panti - is a Grade 12 student at the Philippine Science High School – Main Campus. He is an active participant in the National Olympiad of Informatics Philippines, placing Top 3 in the 2025 IOI Philippine Team Selection and serving as a Teaching Assistant for the NOI Year-Long Training Program (NOI-YTP). He also participated in the International Olympiad of Informatics 2025 in Bolivia.



Farrell Eldrian Wu – is a PhD student in Operations Research at the Massachusetts Institute of Technology, studying information-theoretic algorithms for reinforcement learning and decision making. In high school, he won a gold medal at the IMO and a bronze medal at the IOI. Prior to starting the PhD, he worked in a quantitative research position in industry and as an undergraduate was extensively involved in teaching probability and statistics. He also coaches the Philippines’ IOI 2024 team.

AI in Algorithmic Thinking: Creative Problem Generator

Yahya TABESH¹, Morteza SAGHAFIAN²

¹Plaio Research Group, USA

²Vienna University of Technology, Austria

e-mail: yahya@plaio.world, msaghafi@ac.tuwien.ac.at

Abstract. This study presents a comprehensive architectural roadmap for a purpose-built AI system designed to generate creative, multiple-choice algorithmic and computational thinking problems. Recognizing that effective computational puzzles require narrative logic and visual reasoning rather than rote calculation, the proposed system operates on a “Creativity via Constraints” philosophy. It utilizes a collaborative, multi-agent AI framework—comprising an Architect, an Artist, and a Skeptic—to conceptualize, visualize, and rigorously validate puzzles in a continuous feedback loop.

The core architecture features a curated knowledge base that maps fundamental algorithmic topics, alongside a Chain of Thought (CoT) Generator Module that decouples abstract logic from engaging narrative wrappers to formulate algorithmically plausible distractors. Furthermore, a hybrid Visualizer Simulator Module ensures graphical accuracy on an interactive problem solving platform.

To guarantee and maintain high educational standards, the framework relies heavily on a Human-in-the-Loop (HITL) validation pipeline driven by educator review dashboards, iterative annotator feedback, and rigorous regression testing. Ultimately, this system provides a scalable, reliable tool that augments educators with high-quality, original question sets while immersing students in engaging problem-solving scenarios.

Keywords: algorithmic and computational thinking, creative and interactive problem solving, large language models

1. Introduction

Using an AI model to generate creative algorithmic and computational thinking problems is a uniquely rewarding challenge. Unlike standard exercises, creative computational problems depend on narrative logic, visual reasoning, and satisfying “aha!” moments rather than rote calculation.

A purpose-built AI generator serves two main functions: it immerses students in engaging problem-solving scenarios to enhance their computational thinking, and it provides educators with a continuous stream of original, high-quality question sets. Below is a

comprehensive architectural roadmap for using a system that generates multiple-choice algorithmic puzzles within a hybrid Visualizer Simulator Module to support interactive problem-solving.

2. The Core Philosophy: Creativity via Constraints

To successfully wrap abstract logic in compelling stories and visual puzzles, the generator must operate using three distinct AI “personalities” functioning in a continuous feedback loop:

- The Architect (Generator): Conceptualizes the core algorithmic logic and proposes the narrative wrapper.
- The Artist (Visualizer): Describes or directly generates the visual images needed to support the puzzle.
- The Skeptic (Validator): Actively attempts to solve the generated problem to ensure it has exactly one valid solution, contains no logical leaps, and matches the target difficulty level.

3. System Architecture

A. Data Curation & Knowledge Base: To ensure the AI generates a diverse range of high-quality problems, the system must be grounded in a robust, well-categorized knowledge base. Scrape or manually input existing sets of sophisticated algorithmic topics to serve as a structural body.

- Topics Mapping: The AI must be supported by a comprehensive list of algorithmic topics (e.g., Recursion, Greedy Algorithms, Divide and Conquer, Dynamic Programming, Graph Theory) to ensure broad curriculum coverage.

B. The Generator Module (Prompting Strategy): This module utilizes Chain of Thought (CoT) prompting to decouple the underlying algorithmic logic from the story, ensuring neither is compromised.

- Step 1: The Problem Skeleton. The model first generates the abstract logic and constraints. (Example: “Design a constrained optimization logic puzzle utilizing a greedy algorithm.”)
- Step 2: The Narrative Skin. The model maps the abstract skeleton to an engaging, student-friendly theme. (Example: “Imagine you are at a ‘Fill-Your-Bag’ candy shop, and you want to maximize the total value of the candy you can carry...”)
- Step 3: Intelligent Distractors. Generating the incorrect multiple-choice options (A, B, C, D, E) is just as critical as generating the correct one. The AI must be prompted to calculate plausible errors, such as off-by-one mistakes, partial algorithm completions, or common miscalculations, rather than simply outputting random numbers.

C. The Visualizer Module Algorithmic problems are significantly more intuitive and cognitively engaging when paired with visual aids.

- A hybrid Visualizer Simulator Module ensures graphical accuracy on an interactive problem solving platform.

4. Human-in-the-Loop (HITL) Validation

To maintain a standard of excellence, AI should augment human reviewers, not replace them. The system requires a robust validation pipeline:

- **Review Dashboards:** Educators are presented with candidate problems alongside AI-generated metadata, including a difficulty estimate and specific cognitive/topic tags.
- **Annotator Feedback Loop:** Human judges evaluate and tag the output using categories like “*Too easy*,” “*High quality but too similar to Problem X*,” or “*Brilliant, keep*.” This data is then fed directly back into the system to refine prompts or fine-tune the model.
- **Regression Testing:** Maintain a static suite of “gold standard” problems. Whenever the underlying models or system prompts are updated, the model must successfully solve and correctly categorize this entire suite to ensure system reliability hasn’t degraded.

5. Prompt

5.1 Problem Design

Role: You are an expert computer science teacher for middle or high school students.

Task: Create an engaging, original, and unpublished story-driven multiple-choice question focused on an algorithmic concept, categorized under a randomly selected topic from the provided Algorithmic Topics file or a specific suggested concept.

Constraints:

- **No Jargon:** Do not explicitly name or refer to the algorithmic topic in the story or the question.
- **Age-Appropriate:** The math and logic should be perfectly tailored for the requested grade reading and comprehension level.
- **Engaging Theme:** Bundle the challenge inside a fun, relatable, or adventurous story (e.g., video games, space exploration, mystery solving).

Output Format:

- **The Story/Scenario:** (1-2 short paragraphs)
- **The Question:** (Clear and concise)
- **Answer Options:** (A, B, C, D, E)
- **Answer & Explanation:** (Reveal the correct answer and briefly explain the hidden algorithm in simple terms)

5.2 Illustration

- **Image Generation:** Design a funny image for this problem to give more motivation to the students

5.3 Topics, Keyword, and Ideas

The Topics File includes main algorithmic thinking topics, such as the following:

Binary Search, Greedy Method, Bubble Sort, Inductive Algorithms, Recursive Algorithms, Divide and Conquer, Dynamic Programming, Knapsack Problem, Insertion Sort, Merge Sort, Eulerian Graph, DFS, BFS, Topological Sorting, Decomposition of Graphs, Adjacency Matrix of a Graph, Hamiltonian Tours, Convex Hull, Voronoi Diagrams, Delaunay Triangulation, Tower of Hanoi, Euclid's Algorithm, Josephus Problem, Pascal Triangle, Fibonacci Numbers, Stable Matching, Interval Scheduling, Pigeonholes, Graph Coloring, Latin Squares, Classical Cryptography, Public Key Cryptography, Tiling, Pattern Recognition, Directed Graph, Strongly Connected Components, Decision Tree, K-Nearest Neighbors, Sorting with a Key, Brute-Force, Optimization, Storage Optimization, Dijkstra's Algorithm, Path on Binary Tree, Worst Case Analysis Algorithms, Classification, Game Theory, Boolean Algebra, Pattern Matching, Lexicographic Order, Encoding - Decoding, Diffusion Model, Convolutional Neural Networks, Formal Logic, Triangular Numbers, Approximation, Robotics Algorithms, Queue FIFO, Stack LIFO, Data Representation, Sequencing, Shortest Path, Binary Numbers, Backtracking, State Machine, Minimum Spanning Tree, Network Flow, Parity, Caesar's Cypher, Sudoku, Ruled-Based System, Encryption Decipher, Knots, Turing Machine

6. Platform

To develop algorithmic thinking problems based on our earlier discussions, we can utilize any advanced Large Language Model (LLM) such as Gemini, Claude, or ChatGPT. However, we prefer *EduArena.ai* as a versatile, all-purpose platform. It allows us to generate outputs either with a single model independently or with two models working side-by-side.

Furthermore, we use Canvas for visualization and illustration. The platform also provides interactive problem simulations, enabling users to solve problems directly within an integrated playground.

EduArena.ai is an AI-powered learning and research platform designed for university students to access advanced AI models for free. It offers AI tutors, course-creation tools, and educational chatbots to assist with studies and research, while enabling researchers to test, refine, and improve AI models with real-world feedback.

7. Example


As an example, we upload the Topics file and insert the following prompt to Gemini 3.1 Pro on *EduArena*:

Model Response

<https://www.eduarena.ai/chat/c54aab9b-5507-47f6-a3ad-f52e1fe11372>

The model randomly chose a concept from the topics file, created a problem, incorporated it into a story, and then included an interactive multiple-choice challenge (Figure 1).

🔹 Dragon's Hoard: The Smart Search



Find the hidden diamond in the fewest tries!

The Story

You are playing a new VR game called '**Dragon's Hoard**'. The dragon has hidden its magical diamond inside one of 100 treasure chests lined up in a row. The chests are perfectly ordered and numbered 1 to 100.

The dragon gives you a magical compass to help. When you open a chest, if the diamond is not inside, the compass will either point **LEFT** (meaning the diamond is in a lower-numbered chest) or **RIGHT** (meaning it's in a higher-numbered chest).

The Challenge

If you play perfectly and use the best possible strategy, what is the maximum number of chests you must open to guarantee you find the diamond?

- A) 100
- B) 50
- C) 7
- D) 10
- E) 5

Figure 1. EduArena.ai platform

The platform provided an interactive simulated playground for solving the problem experimentally (Figure 2).

The interface is titled "Interactive Demonstration" and includes a subtitle: "Test the logic yourself! Try to find the diamond manually, or watch the computer use the ultimate halving strategy." Below the subtitle are two buttons: "Reset & Try Yourself" and "Watch Smart Strategy". A dark blue banner contains the instruction "Click any chest to start searching!" and a "Tries: 0 / 7" counter. The main area is a 10x10 grid of orange chests numbered 1 to 100.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Figure 2. Interactive simulated playground

As soon as we select one of the five answer options, the system immediately provides feedback on correctness, along with a detailed explanation.

8. Evaluation

To assess the effectiveness of Large Language Models in generating algorithmic problems, we conducted a side-by-side comparison of Gemini 3.1 Pro and ChatGPT 5.2 Pro using the EduArena platform. The models were prompted to create twenty problems across various specified topics. Table 1 shows the target topics and grade levels from the prompts, along with the raw responses from both models. Tables 2 and 3 display the subsequent performance evaluations for Gemini and ChatGPT, respectively, by human experts.

Table 1. Links to models' response

ID	Topic	Grades	Link to Models' Response
1	Strong Induction	7-8	https://www.eduardena.ai/chat/12f55c2e-2202-47d9-96e0-5338ad15ef8c
2	Greedy Method	5-6	https://www.eduardena.ai/chat/71509aa4-7a1c-496c-86f3-62eb47ce41d4
3	Fibonacci Numbers	5-6	https://www.eduardena.ai/chat/8dfda394-a432-4bdd-abe2-3a678be0af11
4	Latin Squares	7-8	https://www.eduardena.ai/chat/3f09cc3e-5803-434b-93df-6e3b1fdcd8f8
5	Pigeonhole	5-6	https://www.eduardena.ai/chat/17ee4dbd-db21-4150-90e0-99613226d59b
6	Graph Coloring	7-8	https://www.eduardena.ai/chat/9d565222-3962-4e1b-bafd-166d912084e2
7	Directed Graph	5-6	https://www.eduardena.ai/chat/0caa9c3c-68a3-4ae3-a609-1b0c063ea6f0
8	Bipartite Matching	7-8	https://www.eduardena.ai/chat/841fe520-2108-4e9f-8c73-8842030da50b
9	Pascal Triangle	5-6	https://www.eduardena.ai/chat/66e1f9c4-3770-4317-86de-3a2e872c2daf
10	Optimization	7-8	https://www.eduardena.ai/chat/7f48a626-7534-48c3-b3db-6d8366224e14
11	Strong Induction	11-12	https://www.eduardena.ai/chat/ea2aece9-0128-4ca9-90d2-c07eac526d86
12	Greedy Method	9-10	https://www.eduardena.ai/chat/86d63108-5423-4e70-a205-8c2e0cceba54
13	Fibonacci Numbers	11-12	https://www.eduardena.ai/chat/17a7795a-ffde-494b-8269-1554e85b2440
14	Latin Squares	9-10	https://www.eduardena.ai/chat/c22e7977-3e5f-44c9-9bb7-df426d485476
15	Pigeonhole	11-12	https://www.eduardena.ai/chat/43af82ce-80a0-4377-8005-0a935abca86c
16	Graph Coloring	9-10	https://www.eduardena.ai/chat/c2f0124a-5ecc-438b-b11a-97fe3a0977fd
17	Directed Graph	11-12	https://www.eduardena.ai/chat/66bc8c99-1ce8-44dd-ade5-9a3f1146a817
18	Bipartite Matching	9-10	https://www.eduardena.ai/chat/4c5af199-0460-4eea-8304-cc76849c71d5
19	Optimization	9-10	https://www.eduardena.ai/chat/d53f55d6-4aa6-43e0-bf29-c19656ba138a
20	Pascal Triangle	11-12	https://www.eduardena.ai/chat/7d550fec-a953-4f18-a0cf-60fb1153fdd2

Experts evaluated each generated problem in Table 1 based on several criteria: problem correctness and clarity, solution correctness, image accuracy, novelty, appropriate difficulty, topic alignment, and engagement. The evaluation results for Gemini 3.1 Pro and ChatGPT 5.2 Pro are presented in Tables 2 and 3, respectively.

Table 2. Gemini 3.1 Pro

Gemini 3.1 Pro								
ID	Problem Correctness	Problem Clarity	Solution Correctness	Image Accuracy	Novelty	Appropriate Difficulty	Topic Alignment	Engagement
1	Correct	Clear	Correct	Accurate	Not novel	Appropriate	Aligned	Not
2	Correct	Clear	Correct	Accurate	Not novel	Appropriate	Aligned	Somewhat
3	Correct	Clear	Correct	Accurate	Not novel	Appropriate	Aligned	Somewhat
4	Correct	Clear	Correct	Accurate	Novel	Easier	Aligned	Somewhat
5	Correct	Ambiguous	Correct	Accurate	Not novel	Easier	Aligned	Not
6	Correct	Clear	Correct	Accurate	Not novel	Easier	Aligned	Somewhat
7	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Somewhat
8	Correct	Clear	Correct	Accurate	Not novel	Easier	Aligned	Not
9	Correct	Clear	Correct	Accurate	Not novel	Appropriate	Aligned	Somewhat
10	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Engaging
11	Correct	Clear	Correct	Accurate	Not novel	Appropriate	Aligned	Not
12	Correct	Clear	Correct	Accurate	Not novel	Appropriate	Aligned	Not
13	Correct	Clear	Correct	Accurate	Not novel	Easier	Aligned	Not
14	Correct	Clear	Correct	Accurate	Not novel	Easier	Aligned	Not
15	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Somewhat
16	Correct	Clear	Incorrect	Accurate	Novel	Appropriate	Aligned	Not
17	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Somewhat
18	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Engaging
19	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Engaging
20	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Somewhat

Table 3. ChatGPT 5.2 Pro

ChatGPT 5.2 Pro								
ID	Problem Correctness	Problem Clarity	Solution Correctness	Image Accuracy	Novelty	Appropriate Difficulty	Topic Alignment	Engagement
1	Correct	Clear	Incorrect	Inaccurate	Not novel	Appropriate	Aligned	Not
2	Correct	Clear	Correct	Accurate	Not novel	Appropriate	Aligned	Somewhat
3	Correct	Clear	Correct	Accurate	Not novel	Easier	Aligned	Not
4	Correct	Clear	Correct	Accurate	Not novel	Easier	Aligned	Not
5	Correct	Ambiguous	Correct	Accurate	Not novel	Easier	Aligned	Not
6	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Engaging
7	Correct	Clear	Correct	Accurate	Novel	Easier	Aligned	Somewhat
8	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Engaging
9	Correct	Clear	Correct	Accurate	Not novel	Easier	Aligned	Not

10	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Somewhat
11	Correct	Clear	Correct	Accurate	Not novel	Appropriate	Aligned	Not
12	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Engaging
13	Correct	Clear	Correct	Accurate	Not novel	Easier	Aligned	Not
14	Correct	Clear	Correct	Accurate	Not novel	Easier	Aligned	Not
15	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Somewhat
16	Correct	Clear	Correct	Accurate	Not novel	Easier	Aligned	Not
17	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Somewhat
18	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Engaging
19	Correct	Clear	Correct	Accurate	Novel	Appropriate	Aligned	Engaging
20	Correct	Clear	Correct	Inaccurate	Novel	Appropriate	Aligned	Somewhat

Tables 2 and 3 present interesting observations derived from the expert evaluations. We have extracted and visualized these findings in Figure 3, where each cell displays the distribution of scores for a single evaluation criterion.

Although these results stem from comparing two specific LLMs used for problem design, they provide a representative sample of how LLMs generally respond to prompts requesting the design and illustration of specific algorithmic problems.

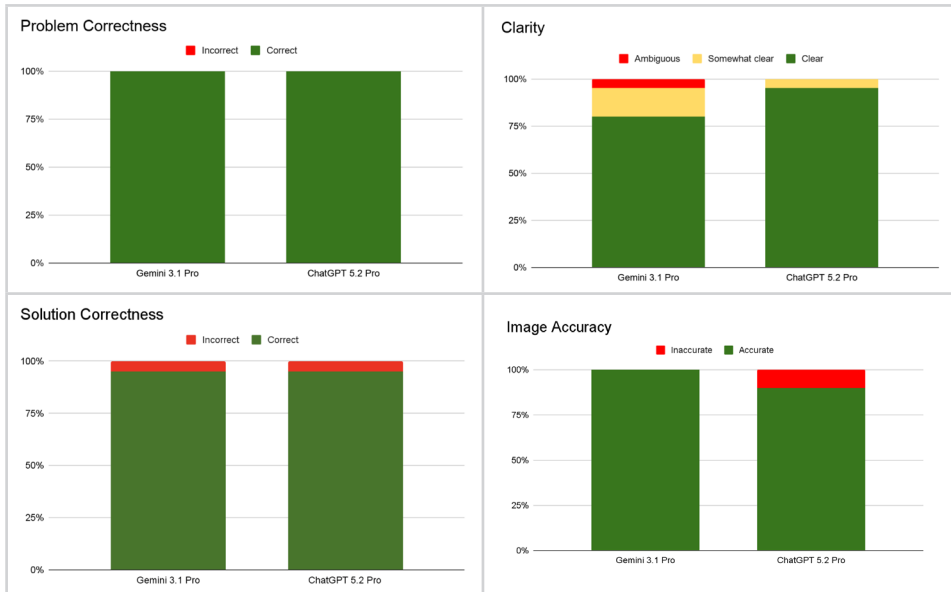




Figure 3. The distribution of scores for a single evaluation criterion

The experts' overall preference for the generated problems is shown in the pie chart below. The two models perform similarly, although Gemini has a slight edge.

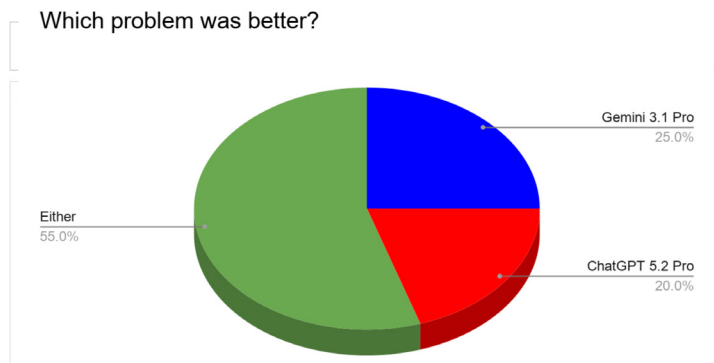


Figure 4. The experts' overall preference for the generated problems

9. Outlook

Effective problem-solving environments offer tools that enable users to create, refine, and control their work, encouraging meaningful creative learning. For young learners, this process must be interactive rather than passive. The main goal is to teach problem-solving strategies while also fostering the joy of creativity.

LLMs can function as an interactive, narrative playground where learners actively build their understanding. Specifically, the interactive algorithmic problem design and solving shown in this work demonstrate the endless resources AI can provide to creative learners. This environment fosters a cognitive style where structured thinking becomes an intuitive and effective learning method. While we have explored only a few ways LLMs can create these problem-solving playgrounds, many more opportunities remain. Moving forward, it is crucial to take advantage of these opportunities to support creative learning through AI, carefully designing systems that amplify human ingenuity rather than replace it.

References

- Kleinberg, J., Tardos, E. (2013). Algorithm Design. Pearson.
- Knuth, D. (1997). The Art of Computer Programming, Vol. 1: Fundamental Algorithms. A-W.
- Google. (2026). Gemini (March 2026) [Large Language Model]. <http://gemini.google.com/>.
- OpenAI. (2026). ChatGPT (March 2026) [Large Language Model]. <http://chatgpt.com/>.
- Turing. (2026). EduArena (March 2026) [Large Language Model]. <http://eduarena.ai/>



Y. Tabesh. Co-founder of Plaio Research Group, Distinguished Emeritus Faculty at the Sharif University of Technology. Co-founder of the Iranian National Informatics Olympiad. Co-founder of the Iranian Mathematical Olympiad. Erdős Award laureate, 2010.



M. Saghafian. Postdoctoral researcher at Vienna University of Technology, working in discrete geometry and computational topology. PhD graduate of Sharif University of Technology. Gold medalist at the International Mathematical Olympiad (IMO). Co-founder of the Iranian Geometry Olympiad and the Iranian Combinatorics Olympiad.

REPORTS

Building a Sustainable Competitive Programming Ecosystem in Argentina: From Olympiad Training to ICPC Institutionalization

Mariano CROSETTI¹, Agustín Santiago GUTIÉRREZ², Micaela BASSANI³, Mariano MALUF⁴

¹ICPC World Finalist & LATAM Champion. ICPC coach. Research Engineer. Applied AI in Aleph (YC S21). Ex-Founder. ex-Google. Member of AAPC's Committee.

²Main Judge at OIA. IOI Silver Medalist. IOI Scientific Committee (Chair, 2025). President of AAPC. MS in Computer Science, University of Buenos Aires. ICPC World Finalist & LATAM Champion. ICPC coach.

³Sponsor Director of AAPC. National University of Rosario, Argentina.

⁴Director of OIA. ECyT, National University of San Martín. IOI Team Leader (2013-2025).

e-mail: marianocrosetti@firo.utm.edu.ar, asgutierrez@dc.uba.ar, micaelabassani.1@gmail.com, mariano.maluf@unsam.edu.ar

Abstract. This paper traces the development of Argentina's competitive programming scene, from high school training through the Argentine Olympiad in Informatics (OIA–IOI) to the establishment of university-level ICPC activities with the founding of the Argentine Association of Competitive Programming. The Argentine experience underscores the importance of educational continuity and community organization in creating sustainable competitive programming ecosystems.

Keywords: ICPC, IOI, OIA, AAPC, Argentina, competitive programming, educational continuity, institutional sustainability, alumni engagement.

1. Introduction: Competitive Programming as an Educational Continuum

Competitive programming offers a structured setting for students to develop algorithmic reasoning, teamwork, and persistence through problem-solving under pressure. Research on computing education increasingly recognizes it as both a cognitive and social learning experience: students strengthen analytical reasoning while acquiring collaboration and communication skills within small teams that mirror real-world software environments (Černý & Mannová, 2011). In this sense, competitive programming functions as a peda-

gological bridge between academic theory and practical computational thinking (Yuen *et al.*, 2023).

In many countries, two distinct competitive programming traditions exist: Olympiad-level training aimed at the International Olympiad in Informatics (IOI), which targets secondary-school students competing individually, and university-level team contests focused on the International Collegiate Programming Contest (ICPC). These traditions generally operate as separate ecosystems with different organizations, formats, and communities. However, they share a common foundation in algorithmic problem-solving, and students who participate in one often go on to compete in the other. The extent to which a country's competitive programming ecosystem supports this transition, and the degree of interaction between the two communities, can have important implications for talent development and long-term sustainability.

Argentina offers a valuable example in this area. The country has run an active secondary-school Olympiad, the Argentine Olympiad in Informatics (OIA), since 1990, and has participated in ICPC since the early 2000s. These two communities have evolved separately: OIA is a government-managed program hosted at a national university, while the ICPC ecosystem began in 1999 and grew naturally through university workshops and volunteer efforts. This growth led to the founding of the Argentine Association of Competitive Programming (AAPC) as an independent nonprofit organization in 2023. The two operate as separate entities with different governance, funding, and activities.

However, tangible connections between them have existed for years and are becoming more apparent. Several individuals serve as coaches or organizers in both communities simultaneously. A number of Argentina's top ICPC performers, including World Finalists, began their competitive programming careers as OIA participants in secondary school. More recently, events like ICPC Training Camp Argentina indicate an emerging, albeit still early, cross-pollination between the two ecosystems involving secondary students. At the institutional level, initial discussions have begun regarding potential integration of initiatives, such as adapting ICPC-style training models for IOI preparation, although no formal coordination structure is currently in place.

This paper traces the parallel development of these two communities and the organic connections that link them. It starts by exploring the foundational role of OIA in introducing secondary school students to algorithmic thinking and international competitions. It then explains how students trained through Olympiad programs carried their skills and motivation into universities, sparking the growth of competitive programming workshops nationwide. Within this ecosystem, the formation of the Argentine Association of Competitive Programming (hereinafter, AAPC) is analyzed as a response to the organizational challenges posed by a volunteer-based model. The paper also discusses Training Camp Argentina (TCA) as a national training platform. It highlights the emerging yet still informal convergence between the OIA and ICPC communities, the tangible impacts of this ecosystem, and the sustainable practices that ensure its future. Finally, the paper offers lessons that could be useful for other national competitive programming ecosystems aiming to strengthen educational continuity across different stages.

2. Early Foundations: The Argentine Olympiad in Informatics (OIA)

The Argentine Olympiad in Informatics (OIA, from the Spanish *Olimpiada Informática Argentina*) has been the primary gateway to competitive programming for high school students in Argentina for nearly thirty years. OIA is a government-supported academic Olympiad; it functions as part of the National Program of Olympiads managed by Argentina's Ministry of Education, which assigns each scientific discipline's Olympiad to a designated national university. In the case of informatics, OIA has been hosted at General San Martín National University (UNSAM) since 1996, when the Olympiad was restarted after a short hiatus following Argentina's early IOI participation in 1990–1994 (Gutiérrez, 2020).

OIA's organizational structure reflects the federal nature of the Argentine educational system. The country is divided into 25 to 30 jurisdictions, covering provinces, Buenos Aires City, and regions within large provinces, each with a coordinator appointed by OIA. The selection process involves three rounds. The Jurisdictional Contest serves as an open, nationwide first round in which any high school student can participate. Since 2019, this round has been conducted as a single, unified contest across the country using the CMS contest management system (Maggiolo & Mascellani, 2012), a major improvement over the previously decentralized selection methods. The National Contest brings together the top performers from all over the country for an onsite competition, and the Selection Contest, designed to mirror the IOI format with two five-hour competition days, determines the four-member team that will represent Argentina at the International Olympiad in Informatics (Gutiérrez, 2020).

Participation data showcase OIA's reach and growth. From 2013 to 2019, the number of students submitting solutions in the Jurisdictional Contest ranged from around 200 to over 460, while the National Contest consistently attracted between 60 and 100 students each year (Gutiérrez, 2020). On the international level, Argentina has participated in over 30 IOI competitions, earning 3 Gold Medals, 9 Silver Medals, 26 Bronze Medals, and 2 Honorable Mentions, demonstrating a strong tradition of competitive excellence at the secondary-school level.

Beyond the competitions themselves, OIA built an ecosystem of educational resources that lowered the barrier to entry for students nationwide. An online judge (OIAJ), based on the social platform, hosts past competition problems through a Spanish-language interface, offering year-round practice opportunities. A curated wiki provides reference materials on algorithms and data structures tailored for secondary-school competitors. Official booklets documenting solutions to all competition problems have been published annually since 2017. Additionally, talks and lectures by former competitors during National Contest events, many of which are recorded and published online, have become an important channel for knowledge transfer between generations of participants (Gutiérrez, 2020).

The educational and cultural foundations built by OIA are important. For many Argentine students, the Olympiad is their first experience with algorithmic problem solving, formal reasoning, and competitive settings. It introduces a teaching method focused on individual effort, increasing difficulty, and repeated improvement of mind habits that can be directly

applied to university-level competitive programming. Likewise, OIA formed a national community of students, teachers, and coordinators united by a shared interest in computational thinking, creating the social infrastructure that would support future developments.

3. Transition to University-Level Competitive Programming

The first Argentine team to gain international recognition came from the University of Buenos Aires (UBA), which earned two Bronze Medals at the ICPC World Finals in Honolulu (2002) and Beverly Hills (2003). The most recent Bronze Medal was secured by the team “InChaVoLa” (UBA) at the ICPC World Finals in Egypt (2023). For more than two decades, UBA remained the only Argentine institution to consistently reach the ICPC World Finals, establishing an early tradition of excellence. However, the arrival of IOI olympiad-trained students at universities across the country gradually shifted this landscape from a single-institution phenomenon to a broader national movement.

At several universities, ICPC coaches started organizing weekly competitive programming workshops. These sessions, usually three hours long and held in university labs, are free and open to students from all STEM fields and experience levels. During these workshops, students receive curated study materials, work on original programming problems, and practice strategies for effective teamwork. Coaches guide participants in problem-solving, time management, and team coordination, combining learning with collaborative exercises that mimic the competitive environment. This hands-on approach strengthens technical skills while encouraging collaboration, creativity, and resilience—key qualities for success in ICPC and beyond.

The expansion of these workshops beyond the University of Buenos Aires was gradual but important. Universities in La Plata (National University of La Plata, UNLP), Córdoba (National University of Córdoba, UNC), Bahía Blanca (National University of the South, UNS), Rosario (National University of Rosario, UNR), and Santa Fe (National Technological University, UTN–Santa Fe), among others, developed their own competitive programming programs. Additionally, over the past few years, ICPC coaches organized online workshops for high school students. This geographic diversification, with participating institutions now spanning more than ten Argentine provinces at the national competition, turned competitive programming into a truly national effort.

A key part of this expansion was the involvement of former contestants who became coaches and organizers, many of whom also took part in IOI. This pattern of leadership continuity fosters an organic system for knowledge transfer across educational levels and institutions. The Argentine model reflects a pedagogical approach aligned with research highlighting the fusion of competition and collaboration (Černý & Mannová, 2011).

4. From Community to Institution: Creation of AAPC

As participation in ICPC-related activities expanded across Argentine universities, the limitations of a purely volunteer-based structure became increasingly evident. The growth

in the number of teams, institutions, and national-level initiatives introduced new demands for coordination, funding, and long-term planning. In this context, the ecosystem required not only committed individuals but also stable organizational mechanisms.

The Argentine Association of Competitive Programming (AAPC), formed in late 2023, is a response to these structural challenges. It was established as an independent nonprofit organization by community members, marking a shift from an informal network to a more organized and official ecosystem.

Its role can be understood through three main functions. First, AAPC provides legal and administrative capacity, enabling the ecosystem to secure funding, formalize sponsorship agreements, and manage resources transparently. This is a necessary condition for scaling activities such as training camps, national contests, and international participation. Second, it centralizes coordination and governance, allowing for the consolidation of previously dispersed initiatives. National competitions, training events, and community activities can now be organized under a shared framework, reducing fragmentation and improving efficiency. This transition from decentralized volunteer efforts to structured coordination has been identified as a critical step in the maturation of competitive programming ecosystems (Blum, 2023). Third, AAPC supports long-term sustainability by reducing dependence on individual organizers. Through a structured governance model and a growing membership base composed largely of former contestants, the association transforms individual commitment into institutional continuity.

Importantly, this process of institutionalization arose from within the community itself. Instead of replacing grassroots dynamics, AAPC formalizes and supports them. In this way, the Argentine case shows how community-driven ecosystems can develop into stable institutions without losing legitimacy or engagement.

5. Training Infrastructure: Training Camp Argentina and National Activities

The consolidation of the ecosystem is closely tied to the development of a nationwide, structured training infrastructure. This infrastructure comprises two main components: recurring local training venues and large-scale national events.

At the local level, universities regularly host workshops and training groups that offer ongoing preparation in algorithms and problem-solving, as mentioned earlier. These spaces make sure learning continues over time and that new participants can gradually join the community.

At the national level, Training Camp Argentina (TCA) and the Argentine Programming Tournament (TAP) form a coordinated annual cycle of preparation and competition. TCA, in particular, has grown into one of the largest training events in Latin America. Recent editions have hosted over 250 participants on-site, an increase from around 80 in its early years (circa 2010), bringing together students from more than 40 universities across 10 Latin American countries. Additionally, the camp's online dissemination has broadened its reach, with official content exceeding 4,000 views on its YouTube platform.

The interaction between these two levels, local continuity and national concentration, is crucial to the ecosystem's function. Local training promotes long-term growth, while national events unify practices, strengthen community bonds, and expose participants to top-level competition.

6. Articulation Between OIA and ICPC

Although OIA and ICPC operate within different institutional frameworks, their interaction is a crucial part of the Argentine ecosystem. The most direct link is the pipeline of participants. Many students who are introduced to competitive programming through OIA continue their development at the university level within ICPC training environments. This reduces the gap usually seen between secondary and higher education. Additionally, there is considerable overlap in community members. Former contestants often participate in both arenas as trainers, judges, or organizers, helping spread knowledge and practices across levels. This interaction also fosters cultural continuity. Shared norms, such as merit-based advancement, disciplined practice, and collaborative learning, persist at both stages, easing transitions and strengthening community identity. The relationship is not one-way. A strong university-level ecosystem increases the value placed on early participation, while alumni involvement boosts secondary-level training.

Taken together, these dynamics imply that OIA and ICPC should be seen not as separate systems but as interconnected parts of a larger ecosystem connected through continuity across educational stages.

7. Growth and Competitive Outcomes

The growth of the Argentine ecosystem can be evaluated through both organizational expansion and competitive achievements.

At the organizational level, competitive programming has shifted from being a highly concentrated activity to a geographically dispersed network. While in the early 2000s, sustained international participation was mainly linked to a single institution (notably UBA), today ICPC teams and training groups are active across universities in more than ten Argentine provinces. Institutions such as UBA, UNLP, UNC, UNR, and UNS, among others, regularly compete in regional contests, indicating a broader, more decentralized foundation of activity.

This expansion is also evident in national-level participation. In 2024, ICPC-qualifying competitions in Argentina, including the Argentine Programming Tournament (TAP), attracted more than 1,200 contestants. Meanwhile, key national programs have shown strong consistency: Training Camp Argentina has been held annually since 2010, and TAP since 2011, maintaining over a decade of continuous activity with increasing participation levels.

At the competitive level, Argentina has consistently dominated the region. From 1999 to 2025, Argentine teams secured the Latin American Championship 15 times (2000, 2002,

2003, 2004, 2007, 2008, 2009, 2011, 2015, 2016, 2019, 2020, 2021, 2023, 2024), accounting for 55.6% of all editions during that period. In other words, Argentina won the majority of Latin American championships in this timeframe.

Argentine universities have also consistently represented at the ICPC World Finals, demonstrating the system's ability to remain competitive internationally. Within this broader ecosystem, the Argentine Olympiad in Informatics (OIA) serves as a key platform for identifying and developing early talent. Operating under a federal structure with nationwide reach, including all 23 provinces and the City of Buenos Aires, OIA provides widespread access to competitive secondary school programming. In recent years, participation in its initial (school-level) stages has increased significantly, with about a 40% growth over the past three years. Beyond rankings or medals, OIA functions as a developmental platform that supports long-term careers in computer science. Many of its former participants continue into university-level competitive programming and related fields, with many later working for leading technology companies. In this way, OIA not only promotes early engagement but also helps sustain the talent pipeline that feeds the ICPC ecosystem.

Beyond quantitative indicators, qualitative outcomes are also important. These include strengthening mentorship networks, fostering greater collaboration between institutions, and increasing recognition of competitive programming as a valuable educational practice within universities.

Together, these elements indicate a system that has reached both scale and performance, backed by ongoing activity, territorial growth, and continuity across educational stages.

8. Conclusions

The Argentine case highlights several factors that support the development of sustainable competitive programming ecosystems. First, maintaining continuity across educational stages reduces attrition and enhances long-term participation. The connection between OIA and ICPC is crucial in this regard. Second, alumni involvement serves to transfer knowledge and sustain organizations, ensuring that experience is retained rather than lost. Third, institutionalization allows for growth and stability without necessarily harming community dynamics, especially when it originates from within the ecosystem. Fourth, the Argentine experience challenges the idea that inclusivity and excellence cannot go hand in hand. In fact, broader access seems to strengthen competitive performance by expanding the talent pool.

These elements are interconnected; their interaction creates systemic effects. The Argentine ecosystem demonstrates how educational, organizational, and community aspects can come together into a unified structure.

References

- Blum, D. (2023). Competitive programming participation rates: an examination of trends in U.S. ICPC regional contests. *Discover Education*. <https://doi.org/10.1007/s44217-023-00034-1>
- Černý, M., & Mannová, B. (2011). Competitive and collaborative approach towards a more effective education in computer science. *Contemporary Educational Technology*, 2(1), 1–9.
- Gutiérrez, A. S. (2020). Argentine Olympiad in Informatics. *Olympiads in Informatics*, 14, 169–176. <https://doi.org/10.15388/ioi.2020.13>
- Yuen, T. T., Liu, L., & Leong, P. (2023). Competitive programming in computational thinking and problem-solving education. *Computer Applications in Engineering Education*, 31(5), 1–15.



Mariano Crosetti is a Research Engineer in Applied AI at Aleph (YC S21). He was a two-time ICPC World Finalist (2015, 2016), winning the Latin America Regional Championship in his last appearance. Since then, he has dedicated significant effort to promoting competitive programming in Argentina: as an ICPC coach at UNR, he has helped teams qualify for seven World Finals from 2017 to 2025; as a problem setter, he contributed problems to the 2019 ICPC Latin America Regional and the 2018–2019 Argentine National Tournament; as an instructor at national and international training camps (Argentina 2018, Colombia 2019, Argentina

2023, Peru 2024); and he led the organization of Training Camp Argentina 2024, which became the largest edition and one of the most recognized events in Latin America for its technical level.



Agustín Santiago Gutiérrez Holds an MS in Computer Science from FCEN-UBA (Buenos Aires University). He is the President of the Argentine Association of Competitive Programming. He has competed twice in the IOI, earning a Silver Medal, and has been a finalist twice at the World and Latin American levels in ICPC, with eight appearances as a world finalist coaching team. Since 2011, he has been actively involved in organizing and managing the Argentine Olympiad in Informatics (OIA) contests. Since 2015, he has served as an official National Jury member

and IOI team trainer at OIA, as well as Deputy Leader of Argentina at IOI. He organized the Iberoamerican Olympiad in Informatics (CIIC) in 2019 and 2020, a regional contest covering Latin America, Spain, and Portugal. He has served as an ISC member at IOI (2022–2025). He has been an organizer and lecturer at most editions of the Argentine ICPC Training Camp and many other camps across Latin America. He was a teaching fellow at

FCEN-UBA from 2013 to 2018, teaching Algorithms and Data Structures. Since 2025, he has been an invited professor at Universidad de San Andrés, teaching Algorithm Design. Additionally, he is a private instructor in Competitive Programming.



Micaela Bassani is the Sponsorship Director at the Argentine Competitive Programming Association (AAPC). She studied Political Science at the National University of Rosario and now leads fundraising and strategic partnerships with tech companies, securing support for training programs and the international participation of Argentine teams at ICPC contests. She is involved in organizing competitions and training initiatives, including Training Camp Argentina 2024-2026 and the Argentine Programming

Tournament (TAP) at UNR (2024–2025). Her work focuses on creating sustainable structures for competitive programming ecosystems.



Mariano Maluf holds a Bachelor's degree in Public Administration from UNSAM and a Degree in Systems Analysis from ORT. He is the Director of the Argentine Olympiad in Informatics (OIA) and has been collaborating with the OIA since 1996. He has been involved in organizing jurisdictional, national, and selective competitions from 2013 to the present. He served as the Argentina Team Leader at the IOI from 2013 to 2025. Currently, he works at the School of Science and Technology at the National University of San Martín.

Developing Computational Thinking for Participation in Programming Olympiads

Michael DOLINSKY

Faculty of Mathematics and Technologies of Programming, F. Skorina Gomel State University,
Sovetskaya str., 104, Gomel. 246019. Republic of Belarus

e-mail: dolinsky@gsu.by

Abstract. This paper examines how to improve schoolchildren’s ability to solve complex programming problems. The author classifies the problems of the International Olympiad in Informatics finals as such. Contrary to the opinion of many schoolchildren and teachers who believe that such skills cannot be developed through training, but are largely determined by hereditary factors, the author took as allies Polya, Altshuller, and Vygotsky, who developed a methodology for the targeted development of thinking. In addition, the article presents the author’s experience in teaching programming and preparing students for successful performance in programming Olympiads, including recommendations for solving problems during Olympiads, developing skills for solving complex problems during training, and using Codeforces for cognitive and psychological training. In addition, the author describes in detail the system of developing thinking in teaching programming to schoolchildren from grades 1 to 11 based on the DL.GSU.BY - an instrumental distance learning system developed under the author’s supervision.

Keywords: computational thinking, international olympiad in informatics, psychology of learning, DL.GSU.BY, Codeforces.com.

1. Introduction

From 1997 to 2024, 17 schoolchildren taught by the author (Dolinsky, 2024a) have won a total of 29 medals at the International Olympiad in Informatics. Of these, 7 are gold, 11 are silver, and 11 are bronze. At the same time, 2 schoolchildren won gold medals, 7 won silver medals, and 8 won bronze medals. It is important to note that the last gold medal was won in 2012, the last silver medal in 2018, and the last bronze medal in 2024. When the last participant was asked why he did not perform better, he answered that he could not come up with complete solutions to the problems. When solving them further, it turned out that he had enough theoretical knowledge to solve for silver, at least. The penultimate bronze winner (in 2021) answered the same question in a similar way and also added that, most likely, the skill of coming up with solutions is not trained but is innate. If we add that in 2024, three of the author’s students won bronze medals, then, naturally, the author had a

question about how to overcome this barrier. This article is precisely the author's attempt to answer this question.

The International Olympiad in Informatics (IOI) was first held in 1989 in Bulgaria. The main goal of the IOI is to promote computer science (CS) among young people and to discover and stimulate young talent in CS. The IOI offers an annual competition in solving algorithmic problems, with solutions implemented in one of several modern programming languages (Verhoeff, 2009). 370 people from 94 countries took part in IOI 2024.

1.1 How to teach problem solving - optimistic foundations

The most famous work in this direction is Polya (1945), which outlines a methodology for teaching problem-solving in mathematics. Polya identifies four stages of problem-solving: understanding the problem, making a plan to solve it, implementing the plan, and analyzing the solution. It is also recommended to develop both heuristic and logical thinking skills. In addition, useful actions at each stage are specified. Particular attention is paid to the psychological aspects of problem-solving, with the final statement that teaching the art of problem-solving is also the education of the will.

Another well-known work in this direction is TRIZ - Theory of Inventive Problem Solving (Altshuller, 2004). This is how the author of TRIZ describes the contents of his book. "The book is intended for a wide range of readers, primarily engineers, developers of new technology, inventors, and students of technical universities. The issues of creative process management are generally considered using inventive examples, so the book is also addressed to readers not involved in technical creativity. The book is of particular interest to scientists and researchers in cybernetics, artificial intelligence, and the psychology of thinking. The principles of thinking management in solving inventive problems (namely, principles, not specific formulas and rules) can apparently be transferred to the organization of creative thinking in any area of human activity."

Vygotsky (1997) is important in teaching schoolchildren how to think. According to Vygotsky, the learning process is understood as a collective activity that occurs in cooperation between an adult and a child. Only that learning is good that anticipates development; that is, learning should precede development and be its source. Then it awakens and brings to life many other functions that lie in the zone of proximal development (ZPD). According to Vygotsky, the ZPD is determined by the content of tasks that the child cannot yet solve independently but can solve in joint activities with an adult. What is initially accessible to the child under the guidance of adults then becomes his own property (skills, abilities).

1.2 Third-party recommendations for learning to solve complex programming problems

The knowledge requirements for participants in the International Olympiad in Informatics are set out in the IOI Syllabus (2025), which aims to determine: the minimum knowledge required for participation in the IOI; whether the proposed task is suitable for the IOI; and the areas of preparation for participants in the IOI. The IOI Syllabus contains the follow-

ing sections: Mathematics; Discrete structures: functions, relations, sets, graphs, and trees; Fundamentals of programming and algorithmization; Data structures; Development, testing, and debugging of programs.

An important tool for preparing schoolchildren for programming olympiads is Codeforces platform (Mirzayanov, 2020), which has been operating since 2010. It regularly hosts personal Olympiads of four difficulty levels. Based on the Olympiad results, the ratings of Olympiad participants are calculated and updated. The author’s analyses are posted for all problems. The authors assign tags to all problems, indicating their topics, and over time, each problem receives a difficulty rating based on the number of people who have solved it. The modes for completing problem solutions, the ability to filter problems by tags and ratings, and virtual contests are supported. The site provides maintenance for users’ personal blogs. As a result, the blogs contain a lot of useful educational and developmental information. The site provides an API for working with the information it collects. Thus, the site, among other things, supports two important tasks: systematic training and assessment of training effectiveness.

In particular, CF ratings can be used to predict results in upcoming Olympiads. For example, the paper *IOI_vs_CF* (2024) provides the following data on the relationship between CF ratings and IOI 2024 medals (Table 1):

Table 1

Relationship between CF ratings and IOI 2024 medals		
Medal	Minimum	Median
Gold	2029	2469
Silver	1501	2274
Bronze	1637	2082

The *IOI-CF-Rating* (2025) paper provides data on the ratings of team participants from countries planning to participate in IOI 2025 (57 countries, 228 participants as of July 1, 2025).

Daniliuk (2018) explains how to read programming task statements. The result of reading the problem statement is usually a pure mathematical model, freed from the problem legend. Mirzayanov (2015) outlined his methods for finding solutions to complex programming problems: “Remember everything”, “From the particular to the general”, “Bold hypothesis”, “To solve a problem, you must think like a problem”, “Let’s think together”, “Choose a method”, “Print and view”, “Google”. Maksimovski (2020) proposed a special strategy for solving problems. Kulkov (2016) collected a variety of techniques for solving non-standard programming problems. When solving the Olympiad, it is important to choose the right strategy for solving a set of problems. Korotkevich (2017), the winner of 7 IOI medals (the best participant in IOI history with 6 gold and 1 silver medals), presented his own strategy. Ray (2024) emphasizes the importance of automated testing to

identify errors. Sahgal (2022) suggests training on problems 200-300 points above your Codeforces rating to develop the skill of solving constructive problems. Luo (2023) gave interesting observations on problems and ways to prepare for their solution. The author gives an unconventional answer to the question “How to improve your Codeforces rating?” - solve math problems.

2. Author’s approaches to solving the problem

2.1 *Work during the Olympiad*

2.1.1 *Recommendations for solving the Olympiad problem set*

Read and understand the conditions of all the tasks. If possible, come up with and implement the simplest (brute force) solutions. Score points for each task. Find out the potential difficulty of each task. Work on each task in order of increasing difficulty as progress is made. If you hit a dead end, switch to another task after a predetermined period.

2.1.2 *Recommendations for solving each problem of the Olympiad*

1. Reading the problem text.

You need to read more slowly, more carefully, more meticulously. DO NOT miss any important facts or comments. After reading, reformulate the condition IN WRITING, leaving the mathematical formulation of the problem. Then, reread the conditions to supplement your problem statement with the omitted facts. Emphasize the facts that should influence the decision. Set the complexity (N , $N\log N$...) of the complete solution.

2. Capturing potential ideas/solution algorithms.

Immediately after reading the conditions, make a list of possible ideas applicable to this task, as complete as possible, so that the right idea is definitely included.

3. Develop a minimal set of tests on which you will check the solution (samples, base case, all possible special/edge cases).

4. Mental rejection of ideas by analyzing their application to a fixed set of tests, and selection of a working idea.

5. Slow, thoughtful coding, ideally eliminating errors in the code.

6. If necessary, built-in control of correct operation (assertions).

Globally

Perhaps you need to spend more time thinking and less time writing (especially debugging).

2.2 Preparation for the Olympiad

2.2.1 Direct development of skills for coming up with solutions to complex problems

Carry out the work described below at least once a week with a gradual increase in the rating of the selected tasks:

1. Choose on Codeforces THREE (like on IOI) problems
 - of your current rating (or 100-200 more)
 - on one of the topics:
 - graphs, complex data structures, dynamic programming, constructive
2. Two hours to think about solutions
3. Read the analyses
4. Write in the relevant DL forum topic in the form
 - link to task
 - task rating
 - result (I came up with it myself or understood the analysis)

2.2.2 Regular participation in rounds on Codeforces

Regular participation in Codeforces rounds brings benefits in the following aspects in preparation for international olympiads in computer science and programming:

- Introduction to modern ideas for solving problems.
- Psychological training in solving problems under time constraints and pressure from the fact that “there is something to lose”, since unsuccessful performance leads to a decrease in the rating on Codeforces, sometimes quite significantly.
- Completing the tasks that were not solved at the Olympiad, from the zone of one’s proximal development (if the author’s analysis is understood).

3. Technology of regular long-term work in the process of preparation for the Olympics

3.1 Distance learning system DL.GSU.BY

The distance learning system DL.GSU.BY (hereinafter, DL, an abbreviation of the phrase Distance Learning) automates assignment issuance, solution checking, table construction for results and ratings, and personalization of learning (Dolinsky, 2022a).

3.2 *Early start*

The simple, intuitive interface of the DL system, as well as special task packages “Learning to work with a mouse”, “Learning to think”, “Learning to read”, “Learning to count”, “Propaedeutics of words”, “Learning words”, provides the opportunity to begin learning programming from the earliest age. In fact, the prerequisites are the ability to walk and talk. In practice, for about 20 years, all students of Secondary School 27 in Gomel have been learning text programming from the 1st grade, subject to their parents’ written application. In addition, preschoolers and students of all schools in Gomel and the Gomel region are also accepted for optional classes.

3.3 *Personalized online learning*

System DL provides personalized learning for each student (Dolinsky & Dolinskaya, 2020). Tasks are issued automatically. If the task cannot be completed, the student can click the “I don’t know” button, which opens the tree-like system of lead-in tasks. Each lead-in task can have its own “I don’t know” button. At the same time, the lead-in tasks include an “I understand” button that lets you quickly return to the task that caused the problem.

In fact, a student can study independently both at school and at home.

3.4 *Specialized package of tasks “Learning to think”*

The tasks were developed under the author’s guidance on the development of the following basic mental operations, grouped into 5 groups:

- Operations on pairs: comparison, ordering, association.
- Operations on sets: union, intersection, subtraction.
- Operations on a set: classification, structuring, generalization.
- Logical operations: negation, conjunction, disjunction, equivalence, and implication.
- Complex operations: synthesis, memorization, analysis, imagination, analogy, abstraction, positioning.

The “Learning to Think” task package (Dolinsky, 2014) consistently develops the skills of all these thinking operations, while the student is offered tasks in the form of drawings, and as the tasks are completed, you need to click in the right places or move the drawings using the mouse or the “Arrow” keys.

3.5 *End-to-end targeted development of thinking in the process of learning programming*

In the course “Informatics”, primary school students undergo sequential personalized programming training in the topics of “Number”, “Char”, “String”, “String Length”, “Posi-

tion of Char in String”, “built-in Delete procedure”, “built-in Copy function”, “built-in Pos function”. One-dimensional array. Two-dimensional array. Geometry. Strings. Sorting. Text problems.

The training is structured in such a way. At the beginning, a task is given. If the student can solve the task, he solves it; if not, he presses the “I don’t know” button and enters the task package, which gradually teaches him how to solve it. At the same time, the main principle is always maintained: first the task, and only if the student cannot complete it independently does he receive an explanatory task. That is, the student is constantly pushed to think. If you can come up with it yourself, you will move on faster.

In addition to the tasks of teaching writing programs, there are regularly interspersed tasks that develop the above-mentioned basic thinking operations, but based on fragments of programs.

3.6 Parallel development of mathematical problem-solving skills

In addition to the course “Informatics” (where students in grades 1-4 learn programming) and “Basic Programming” (where students in grades 5-11 learn programming), there is a course called “Mathematics”. It offers math problems. There is no training or explanation. There are only problem conditions, and you are required to either enter an answer or choose an answer from several presented. An important advantage is the huge number of problems of varying complexity from various sources. Including:

- Kangaroo (2001-2024) for grades 1-10 (in Russian)
- Beaver (2013-2018) for grades 1-11 (in Russian)
- Kangaroo (Canada, Pakistan, Portugal, United Kingdom)
- Beaver (Australia, Canada, Pakistan, United Kingdom)
- Canadian Math Contests
- United Kingdom Math Contests
- Texas University Math Contests

In addition, for grades 1-4, packages of tasks have been completed that accurately monitor the acquisition of knowledge in the mathematics program. The topics and subtopics for which tasks have been consistently prepared for all grades are listed below: 10 tasks for each subtopic.

1st grade

1. Comparison. Space. Time: Identification and difference by 1 feature. Mutual arrangement of objects. Time concepts. Classification by 1 feature. Classification by 2 features. Ordering. Comparison. Naming numbers from 1 to 10. Naming numbers from 11 to 20. Counting objects
2. Single-digit numbers: Numbers from 0 to 10. Addition and subtraction up to 10. More, less, equal. Solving simple problems
3. Two-digit numbers: Numbers from 11 to 20. Addition and subtraction up to 20.

Point, line, curve, segment. Centimeter, decimeter. Hour

2nd grade

Addition. Subtraction. Numbers from 1 to 100. Equations and inequalities with + and -. Right angle, rectangle, square. Broken line. Perimeter of triangle, quadrilateral, rectangle, square.

3rd grade

Multiplication. Division. Equations and inequalities with multiplication and division. Speed. Simple problems on motion. Ray, types of triangles. Three/four-digit numbers. Meter, kilometer. Gram, kilogram, ton.

4th grade

Numbers up to a million. Addition of three- and four-digit numbers. Subtraction of three- and four-digit numbers. Multiplication. Division. Length: mm, cm, dm, m, km. Weight: ton, centner, kg, gram. Time: century, year, month, week, day, hour, minute, second. Area.

The tasks are based on flash technology, so they can be completed in many different ways, including: transferring a drawing to a specific area of the screen, connecting pairs of dots, indicating the appropriate drawing, coloring specific areas of the drawings in the desired colors, choosing the correct answer from the suggested list of options, transferring words to specific areas of the screen, entering the desired numbers in the specified input fields.

A note about the obsolescence of flash technology. The flash task designer has been developed since 2010, and a huge number (thousands) of educational and control tasks have been made with its help. Currently, flash technology is not supported by many browsers by default. At the same time, to use flash tasks, there is an option to enable flash support, which is available in classroom browsers and on schoolchildren's computers. For security reasons, students are advised to work only on DL in such browsers.

3.7 Study of algorithms and data structures

Teaching algorithms and data structures is focused on students mastering the IOI Syllabus.

The course "Basic programming" contains a package of assignments "Accelerated course - 2013", which contains assignments on the topics:

1. "Introduction to Programming" includes tasks that require entering initial data and outputting the answer, i.e., to solve them, you don't need to know anything except input and output operators. At the same time, these tasks serve as propaedeutics for knowledge that may be needed in the future, including mathematics, programming languages, and competitive programming. At the moment, the topic "Introduction to programming" contains tasks on the following subtopics: formatted output; algebraic formulas; numerical operations (AND, OR, XOR, DIV, MOD, SHL, SHR); built-in functions and procedures (ABS, SQR, Odd ROUND, TRUNC, ORD, CHR, UPCASE, STR, VAL, LENGTH, COPY, DELETE, INSERT, POS); number systems.

2. “One-dimensional array” includes tasks for solving which it is necessary to know the declaration of a one-dimensional array, the IF condition operators, and the FOR and WHILE cycles. At the same time, it contains tasks on both standard algorithms for processing one-dimensional arrays and tasks on the propaedeutics of useful knowledge. Currently, the topic “One-dimensional array” contains tasks on the following subtopics: sum; counting; maximum/minimum; maximum/minimum number; cycle range; even/odd positions; long arithmetic (addition, subtraction, multiplication of a number by a digit, divisibility signs), prefix/suffix sums, maximums, minimums; counting sort; cycle parameters; cyclic counting; dec; search; consecutive; all different.
3. “Two-dimensional array” includes tasks for solving which it is additionally required to know the declaration of a two-dimensional array and apply nested cycles. Currently, the topic “Two-dimensional array” contains tasks on the following subtopics: subarray - sum; subarray - counting; counting by rows; array generation; array modification; counting by array perimeter; prefix sums, maximums, minimums; comparison of array rows.
4. “Geometry” includes problems that require the ability to find the distance between two points, a point and a set of points, between all points of a set, and then apply previously studied algorithms on one-dimensional and two-dimensional arrays. In addition, this topic includes problems on basic geometric concepts such as perimeter and area. Currently, the Geometry topic contains problems on the following subtopics: rectangle, Manhattan distance, distances from one point to several points, distance between identical points of two sets, adjacent distances, distances between all pairs of points, distances between all pairs of points of two sets, and area of a polygon.
5. “Strings” includes tasks that require knowledge of the data types symbol, string, string array, and the ability to invent and debug your own algorithms. In fact, this topic is key to assessing the student’s potential. Currently, the “Strings” topic contains tasks on the following subtopics: cyclic shift to the right; if; reversing a string; counting in a string; maximum in a string; Ord; searching in a string; bracket strings; lengths of array strings; counting in an array of strings; generating a string; generating an array of strings; converting a sentence to an array of words; forming arrays of strings; subarray of characters; array of strings - editor; analysis of all cyclic shifts of a string.
6. “Sorting” includes tasks on the ability to apply the algorithm of sorting by exchange, bubble, counting, and includes tasks on the following subtopics: only sorting; fixed number; fixed numbers; post-condition; variable numbers; variable range of numbers; sorting with numbers; sorting by counting; compression of coordinates; all different in ascending order.
7. “Text Problem” contains problems with original texts of problem conditions from the Belarusian Republican or Regional Olympiad (usually one or two pages), in which the task is modified in such a way that its solution does not require knowl-

edge greater than that used to solve problems from topics 1-6. The main difficulty for the participant is to extract the algorithmic formulation of the problem from the text condition. At the moment, Topic 7 includes subtasks on the following algorithmic subtopics: one-dimensional array: sum, sum + condition, counting, maximum, maximum number, minimum, minimum number, conditional sum, conditional minimum, element selection, counting sort, divisors, condition, cycles; two-dimensional array: maximum number in column, row sum, maximum in array, adjacency matrix; row array: counting by array; counting by rows.

8. “Elements of Number Theory” includes both digit-searching problems and problems requiring preliminary study of the relevant theory. Currently, the topic “Elements of Number Theory” includes subtasks on the following topics: For loop; nested For loops; While loop; For + While; divisors; primality testing by definition; Sieve of Eratosthenes; number systems; bit processing; submasks.

Also in the course “Basic programming”, there is a package of tasks “Olympiads 9-11 grades”, which contains tasks on the following topics:

9. “The Greedy Algorithm” includes problems that require pre-sorting of the input data. Currently, this topic includes problems on the following subtopics: Quadratic sorting; Sort with numbers; Counting sorting; Quicksort; Quicksort + while; Quicksort with numbers; Comparison function; 1 2 3 sorting; Correct bracket sequence; Maximum depth of bracket expression; Two arrays; Bid selection; Deadline and price; Minimum cover; Brute force + greedy; Stack + greedy; Ad hoc.
10. “Queue” includes tasks that require knowledge of the theory on the topic to solve. Currently, this topic contains tasks on the following subtopics: knight; labyrinth; three-dimensional labyrinth; pieces; three-dimensional pieces; knight with a dynamic list of moves; numerical sequences; queue exploration; dec; 01-BFS; two queues; bit processing.
11. “Recursion” includes tasks that require using a recursive procedure or function call to solve the problem. Currently, the Recursion topic contains tasks on the following subtopics. The set of all subsets: output of one way to make a sum M , output of all ways to make a sum M , number of ways to make a sum M , number of ways to make a sum not less than M , maximum sum not exceeding M , minimum excess of a sum, subset with the maximum number of suitable elements, forbidden subsets, sum of K subsets. Combinations: quantity, output. Placements. Permutations. Permutations with repetitions. Bracket expressions. Gray code. Fast exponentiation. Number generation. By definition. All subsets of rows of a two-dimensional array. On a two-dimensional array. Divide and conquer. Recursion with memoization. Recursion with memoization by profile.
12. “Dynamic programming and recurrence relations” includes problems of the subtopic: one-dimensional array: all sums, subsequence of maximum length, number of subsequences of maximum length, subsequence of maximum length in $O(N \cdot \log N)$, knapsack, sum of several previous, sum of several previous with answer recovery, maximum of sums, partition into subarrays, prefix sums, prefix

maxima, suffix minima, permutation-number; two-dimensional array: sum of several previous, maximum of several previous, minimum of sum of several previous, sum of maxima of several previous, maximum frame, prefix sums.

13. “Graphs” includes tasks that require graph analysis and processing and contains the following subtopics. Vertex enumeration. Vertex degree. Adjacency matrix. Queue: checking a graph for bipartiteness, shortest paths (Dijkstra and Floyd algorithms), articulation points. Recursion: path with maximum number of edges, reachability matrix, sources and sinks, vertex reachability, unreachable vertices, connectivity, connected components, strongly connected components, dominance sets, finding cycles, Euler cycle, Hamiltonian cycle, path in a directed graph, topological sorting, maximum matching, Kuhn’s algorithm. Definition of a tree, number of edges on the path to a vertex, tree diameter, least common ancestor, order of visiting vertices by depth-first search, centroid decomposition, Huffman character encoding, binary tree, quaternary tree. Minimum spanning tree: Prim and Kruskal algorithms. Disjoint set systems. Strategy games. Hidden graphs. Euler formula. Maximum flow.
14. “Complex data structures” includes tasks that require theoretical knowledge of the following topics to solve. Segment tree: without modification (maximum, minimum, sum, maximum of sums); single assignment: sum, minimum, maximum, minimum segment where there are all numbers from 1 to K); single increment (sum); increment on segment directly, increment on segment lazy propagation (sum, minimum, maximum, number of positives, access to element); assignment on segment (sum, number of segments of ones). Fenwick tree. Trie. Bit trie. Search tree. Suffix array.
15. “Complex Dynamic programming” includes tasks, the solution of which requires knowledge of the relevant theory and the ability to come up with a way to apply it to the following subtopics: DP by bit masks, DP on a tree, Binary lifting, DP by profile, DP on strings, DP on number, DP on bit number, inclusion-exclusion principle.

The tasks in the “Accelerated Course 2013” and “Olympiads 9-11 grades” packages are supplemented with tasks from regional programming olympiads.

3.8 *Weekly training olympiads for beginners*

To control the quality of teaching by students and teachers, DL holds weekly training Olympiads for beginners. Each of them contains 20 problems. The first 10 are on the topic of “Introduction to programming”; the next 5 are on the topic of one-dimensional arrays; the next 5 are one each on the topics of “Two-dimensional array”, “Geometry”, “Strings”, “Sorting”, and “Text problem”. The problems are available for solution during the week, since they open on Thursday morning and close on the following Wednesday evening. Regional Olympiads for students in grades 1-4 of previous years are used as training Olympiads (Dolinsky, 2022b).

3.9 Weekly training olympiads for professionals

To control the quality of teaching by students and teachers, DL holds weekly training Olympiads for professionals. Each of them contains 15 problems on the topics listed above, 1-15. The problems are available for solution during the week, as they open on Thursday morning and close on the following Wednesday evening. Regional Olympiads for students in grades 5-11 of previous years are used as training Olympiads (Dolinsky, 2024b).

3.10 Regional programming olympiads for primary and secondary schools

During the academic year, 5 regional Olympiads are held for students in grades 1-4 and grades 5-11 (two in the fall and three in the spring). To ensure continuity of education, since 2016, the last 5 problems of the Olympiad for grades 1-4 have been identical to problems 3-7 of the Olympiad for grades 5-11. The problems and tests for them for all Olympiads are compiled by the author based on the following considerations. Each academic year, the best-prepared schoolchildren participate in specialized classes to prepare for regional, republican, and international Olympiads. Within the framework of these classes, problems from international and national Olympiads are solved every Sunday from 9.00 to 14.00. Upon completion of the solution, a discussion of the solution to the problems is held. During this discussion, problems on topics/subtopics that are not yet in the “Olympiads 9-11” problem set emerge. Such problems are sent to the “task bank” for use in subsequent regional Olympiads for grades 5-11 on topics 8-15. And problems on topics 1-7 are made as preludes to problems on topics 8-15.

3.11 Specialized classes to prepare for regional, national, and international Olympiads

As already mentioned above, training Olympiads are held every Sunday, close to the real Olympiads. From September to the end of November, team Olympiad problems are solved in teams of 3 students per computer in order to prepare for the Open All-Russian Team Olympiad of Schoolchildren in Programming. From December to the end of March, personal Olympiad problems are solved to prepare for the Gomel Regional and the Belarusian Republican Olympiads of Schoolchildren in Informatics. From April to the end of August, team Olympiad problems are solved again. However, as in personal Olympiads, the problems are programmed by each student independently on their own computer. At the same time, as in team Olympiads, it is allowed to discuss solutions with other training participants. In addition, weekly on Wednesdays, problems are solved in classes. It is clear that the problems can also be solved at home at any time after the Olympiad.

3.12 Seasonal Cups and Person of the Year

To increase schoolchildren’s motivation to do their homework, awards are given quarterly to students who have solved the most problems (main problems; problems from “I don’t know” are not taken into account). Students of grades 1-4 – in the “Informatics” course

(the three best + the four best from Gomel). Students of grades 5-8 in the “Basic Programming” course (the three best). Students of grades 1-8 in the “Mathematics” course (the three best). In the “Programming – Professionals (teams)” course, awards are given to 3 teams of 3 students for the fall, 9 best students for the spring and summer, as well as 9 participants who scored the most points in the winter in the above-described Sunday personal Olympiads in specialized classes.

In addition, once a school year at the beginning of September, the winner of the “Person of the Year” competition will be awarded in the following nominations: “Informatics”, “Basic Programming”, “Mathematics” – one person each who has solved the most problems in the corresponding course over the past academic year.

It is interesting to note that the award ceremony is being held by a Belarusian company OpenMyGame, founded by graduates of the above-described educational system, former winners of the Belarusian Republican Olympiads in Informatics.

3.13 Tracking the dynamics of ratings on Codeforces

To increase motivation, a system for automatically updating the Codeforces ratings table of Gomel and Gomel region students (Gomel_Codeforces, 2025) has been developed within the DL system. Each student sees not only their rating but also their place in the hierarchy of schoolchildren in Gomel and the Gomel region. The first 25 in Gomel qualify for the Gomel Regional Olympiad in Informatics. The first 25 in the Gomel region qualify for diplomas of the Gomel Regional Olympiad. The first 15 in the Gomel region qualify for participation in the Belarusian Republican Olympiad in Informatics. The first 11 people are eligible to receive a diploma of the Belarusian Republican Olympiad in Informatics - from 2018-2025, every year, 11 to 14 representatives of the Gomel region received diplomas of the Belarusian Republican Olympiad in Informatics and automatically received the right to enter any university in Belarus without exams.

3.14 Participation in official Olympiads

In October, the Gomel qualifying Olympiad for the Open All-Russian Team Olympiad of Schoolchildren in Programming (VKOSHP) is held, after which several Gomel teams advance to the VKOSHP finals. In 2023 and 2024, Gomel teams won VKOSHP medals.

In December, the Gomel City Olympiad of schoolchildren in informatics is held; in January, the Gomel Regional Olympiad; in March, the Belarusian Republican Olympiad. In 2023-2025, representatives of Gomel (the author’s pupils) were selected for the Belarusian team at the International Olympiad in Informatics.

4. Conclusion

This paper examines how to improve schoolchildren’s ability to solve complex programming problems. The author classifies the problems of the International Olympiad in Infor-

matics finals as such. Contrary to the opinion of many schoolchildren and teachers that such skills cannot be developed through training but are largely determined by hereditary factors, the author took as allies Polya, Altshuller, and Vygotsky, who, in their works, sought to develop a methodology for the targeted development of thinking. In addition, the article presents the author's experience teaching programming to schoolchildren and preparing them for successful performance in programming Olympiads, including recommendations for solving problems during Olympiads, developing skills for solving complex problems during training, and using Codeforces for cognitive and psychological training. In addition, the author's programming-teaching system from grades 1 to 11 is based on the DL.GSU.BY distance learning tool developed under the author's supervision is described in detail.

The paper was written in spring 2025. But in summer 2025, in Bolivia, two of the authors' pupils won medals: silver and bronze. So now statistics have changed: From 1997 to 2025, 17 schoolchildren taught by the author (Dolinsky, 2024a) have won a total of 31 medals at the International Olympiad in Informatics. Of these, 7 are gold, 12 are silver, and 12 are bronze. At the same time, 2 schoolchildren won gold medals, 8 - silver medals, 7 - bronze. It is important to note that the last gold medal was won in 2012, the last silver medal in 2025, and the last bronze medal in 2025.

From the review: "What I missed in the paper: a more accurate description of the number of people involved in this project and their workload, which seems enormous if it has to be done regularly by one or even a few people."

It seems enormous, but it is so.

The author has been teaching computer science to schoolchildren since 1993. Since 1999, the author has been leading the development of DL.GSU.BY, being a teacher at Gomel State University named after Francisk Skorina. All software is developed by schoolchildren, students, and postgraduates. Problems for olympiads of grades 1-4 are prepared by informatics teachers at Gomel School 27 (ideas) and the author (implementations). Problems for olympiads for grades 5-11 until 2016 were prepared by students and postgraduates; since 2016, they have been prepared by the author. Problems for weekly olympiads are prepared by the author. Math problems is prepared by students.

Classes for students in grades 1-4 of Secondary School 27 are taught by teachers from Secondary School 27. Classes for students in grades 1-4 of other schools, as well as for students in grades 5-11 of Secondary School 27 and other schools in Gomel and the surrounding region, are held weekly by the author on Wednesdays and Sundays.

Also, since the DL site is accessible on the Internet, any student can study on their own and any teacher can teach their students on it. Now more than 100,000 persons from more than 100 countries are registered at DL.

References

- Altshuller H. (1994). *The Art of Inventing (And Suddenly the Inventor Appeared)*. Translated by Lev Shulyak. Technical Innovation Center, Worcester, MA, 1994. ISBN: 0-9640740-1-X.
- Daniliuk A. (2018). How to read problem statements. <https://codeforces.com/blog/entry/62730>
- Dolinsky, M. (2014). Technology for the development of thinking of preschool children and primary school children. *Olympiads in Informatics*, 8, 63–68. https://ioinformatics.org/journal/v8_2014_63_68.pdf
- Dolinsky, M. (2022a). Instrumental System of Distance Learning DL.GSU.BY and Examples of its Application. *Global Journal of Computer Science and Technology Interdisciplinary*, 2022. 22(1):45-53. https://globaljournals.org/GJCST_Volume22/6-Instrumental-System-of-Distance-Learning.pdf
- Dolinsky, M. (2022b). Primary School Programming Olympiads in Gomel Region (Belarus). *Olympiads in Informatics*, vol. 16, pp. 107-123, 2022. https://ioinformatics.org/journal/v16_2022_107_123.pdf
- Dolinsky, M. (2024a). Teaching Programming to Schoolchildren in Gomel (Belarus). *WSEAS Transactions on Advances in Engineering Education*, vol. 21, pp. 11-16, 2024. <https://wseas.com/journals/articles.php?id=9134>
- Dolinsky, M. (2024b). High School Programming Olympiads in Gomel Region. *Olympiads in Informatics*, vol. 18, pp. 155- 166, 2024. https://ioinformatics.org/journal/v18_2024_155_166_Dolinsky.pdf
- Dolinsky, M., Dolinskaya, M. (2020). The Technology of Differentiated Instruction in Text Programming in Elementary School Based on the Website dl.gsu.by, *Olympiads in Informatics*, 14, 37-46. https://ioinformatics.org/journal/v14_2020_37_46.pdf
- Gomel_Codeforces (2025). Codeforces Ratings of the Gomel region schoolchildren. <https://dl.gsu.by/codeforces/>
- Holyinq A. (2012). Sports programming and cool ideas. (In Russ). <https://codeforces.com/blog/entry/5990>
- IOI Syllabus (2025). <https://ioinformatics.org/files/ioi-syllabus-2025.pdf>
- IOI vs CF performance: 2024 Edition (2024). <https://codeforces.com/blog/entry/134654>
- IOI 2025 Teams CF ratings. <https://codeforces.com/blog/entry/142475>
- Korotkevich, G. (2017). My strategy at AtCoder & CS Academy. <https://codeforces.com/blog/entry/53457>
- Kulkov, A. (2016). General ideas for solving non-standard problems. (In Russ). <https://codeforces.com/blog/entry/48417>
- Luo, Z. (2023). The Reason You are Bad at Codeforces — You are Not Russian Enough. <https://codeforces.com/blog/entry/126310>
- Maksimovski V. (2020). Tips on Constructive Algorithms. <https://codeforces.com/blog/entry/80317>
- Mirzayanov, M. (2015). How to Come Up with Solutions to Problems: Techniques. <https://codeforces.com/blog/entry/20548>
- Mirzayanov, M., Pavlova, O., Mavrin, P., Melnikov, R., Plotnikov, A., Parfenov, V., Stankevich A. (2020). Codeforces as an Educational Platform for Learning Programming in Digitalization. *Olympiads in Informatics*, 2020, Vol. 14, 133–142. https://ioinformatics.org/journal/v14_2020_133_142.pdf

- Polya, G. (1945). How to solve it. Princeton University Press, 148 p. <https://math.hawaii.edu/home/pdf/putnam/PolyaHowToSolveIt.pdf>
- Ray, S. (2024). How to Solve Questions [Dominater Version]. <https://codeforces.com/blog/entry/133289>
- Sahgal, S. (2022). How to improve my Constructive Algorithm Thinking? (In comments). <https://codeforces.com/blog/entry/107782>
- Verhoeff, T. (2009). 20 Years of IOI Competition Tasks. Olympiads in Informatics, 2009, Vol. 3, 149–166. <https://ioinformatics.org/journal/INFOL047.pdf>
- Vygotsky, L. S. (1997). Educational Psychology. CRC Press, 416 p. <https://doi.org/10.4324/9780429273070>



Michael Dolinsky is a lecturer in Gomel State University “Fr. Skoryna” from 1993. Since 1999 he is a leading developer of the educational site of the University (dl.gsu.by). Since 1997 he is heading preparation of the scholars in Gomel to participate in programming contests and Olympiad in informatics. He was a deputy leader of the team of Belarus for IOI’2006, IOI’2007, IOI’2008 and IOI’2009. His PhD is devoted to the tools for digital system design. His current research is in teaching Computer Science and Mathematics from early age.

Open Contests in the Japanese Olympiad in Informatics

Tetsushi ITO¹, Seiji HIRATA², Kengo SHIMIZU², Hirotaka YONEDA²

¹Kyoto University

²The University of Tokyo

e-mail: tetsushi.ito@ioi-jp.org; ret2home@gmail.com; kiyomizu@is.s.u-tokyo.ac.jp; squar37@gmail.com

Abstract. The Japanese Committee for the International Olympiad in Informatics (JCIOI) has organized open contests since 2013. These contests were originally based on tasks and contest environments prepared for the Japanese Olympiad in Informatics (JOI), but were made open to the public to contribute to the global competitive programming and informatics education communities. In this article, we describe the format, history, tasks, and technical aspects of open contests organized by JCIOI.

Keywords: open contest, online programming contests, contest management system.

1. Introduction

The Japanese Committee for the International Olympiad in Informatics (JCIOI) organizes the Japanese Olympiad in Informatics (JOI) and selects the Japanese delegation for the IOI. In addition, JCIOI has organized Open Contests since 2013.

The primary purposes of JOI are to encourage and train talented students in informatics and to select the Japanese delegation for the IOI. However, the tasks and contest environments prepared by JCIOI can also be valuable resources for people outside Japan. For this reason, JCIOI has made some of its contests open to participants worldwide. Some of these activities are also supported by sponsors and the Japanese government. We regard the Open Contests not only as part of our training and selection activities but also as a way to contribute to the global competitive programming and informatics education communities.

In this article, we describe JCIOI’s activities related to Open Contests, including their format, history, tasks, and technical aspects. We hope this article serves as a useful resource for those interested in JCIOI’s Open Contests.

2. Overview of the Contest Format

In this article, we use the term “Open Contest” to refer to an online programming contest in which participation is not restricted and anyone may take part freely. JCIOI has organized the following two types of Open Contests.

- (Type 1) Contests using original tasks are held under the name “JOI Open Contest”. These contests were originally designed as training opportunities for members of the Japanese delegation for the IOI. We made them open to the public so that participants from other countries could also benefit from the contests. The JOI Open Contest was held from 2013 to 2025.
- (Type 2) Contests are held at almost the same time as official JOI contests. These contests use the same tasks as the corresponding official JOI contests. They are also called “mirror contests” of JOI.

Both types of Open Contests are based on tasks and contest environments prepared primarily for JOI. By opening them to the public, JCIOI aims to make these resources available to the competitive programming and informatics education communities worldwide.

In what follows, we refer to both types of contests as “Open Contests”.

In addition to these contests, JCIOI also makes judges for past JOI tasks available on the AtCoder website: <https://atcoder.jp/contests/archive?category=200>

These are open to anyone, but they are not discussed in this article.

The contests in (Type 1) and (Type 2) differ only in the source of the tasks. Their operation as Open Contests is essentially the same. However, in (Type 2), JCIOI prioritizes the official JOI contest held at the same time. Therefore, responses to technical or other issues in the Open Contest may sometimes be delayed.

The general flow of an Open Contest is as follows. Details may vary slightly from year to year. For more information, see: <https://contests.ioi-jp.org/>

We announce the Open Contests through several channels, including the IOI-announce mailing list and Codeforces blog posts. Some contestants also share contest information on X (formerly Twitter). These announcements have been important for reaching participants outside Japan.

The basic format is the same as IOI. The contest lasts 5 hours and includes 3 tasks. The contest system is CMS. Usually, the only available programming language is C++. This is because the Japanese IOI delegation members use only C++, and one of the main purposes of Open Contests is to provide training opportunities for them.

When organizing an Open Contest, we prepare the contest tasks and the judging environment. The tasks may be original (Type 1) or drawn from another official JOI contest (Type 2). In either case, to encourage participation from around the world, the task statements are translated into English, and the contest is conducted bilingually in Japanese and English. Recent advances in machine translation have helped reduce the cost of preparing English statements, but we still carefully review them to ensure clarity and precision.

The contest website also provides an account registration page. Participants create accounts and then take part in the contest. In the past, the contest time slot was fixed. Because participants are in many different time zones, we introduced a system that lets each participant choose their own start time. Regardless of when a participant starts, the contest lasts 5

hours. However, the five-hour period must fit within the prescribed contest window. If the overall Open Contest window ends during a participant's contest, that participant's contest ends at that time, even if five hours have not yet elapsed.

Source code submitted during the contest is compiled, executed, judged, and scored in the judging environment, and feedback is returned to the participant. This is the same as in ordinary programming contests such as the IOI.

During the contest, participants may submit questions through the contest system. However, for operational reasons, it may not always be possible to answer them immediately.

After the Open Contest ends, the official input and output data are released, and the contest enters Analysis mode. During Analysis mode, participants may resubmit their solution programs.

The real-time standings are available during the contest. Participants who do not wish to be influenced by others' scores are advised not to view the standings page. After the contest ends, the standings are frozen and published.

No prizes are awarded to participants or to high-ranking contestants.

The rules for Open Contests are kept to a minimum. The 24-hour contest window introduced in 2020 has made the contest more accessible to participants in different time zones, but there is a risk that task information may be shared before some participants start, and it is difficult for the organizers to answer questions throughout the entire contest window. The operation of Open Contests necessarily relies on participants' integrity.

3. History of the JOI Open Contest

The following is a list of past JOI Open Contests. All of these used original tasks, corresponding to Type 1 in the previous section.

For the tasks, data sets, and standings, please see the following website: <https://contests.ioi-jp.org/>

JOI Open Contest, 2013

Sunday, June 23, 2013, 04:00–09:00 UTC/GMT

JOI Open Contest, 2014

Day 1, Round 1: Sunday, June 22, 2014, 04:00–09:00 UTC/GMT

Day 1, Round 2: Sunday, June 22, 2014, 10:00–15:00 UTC/GMT

The tasks for Round 1 and Round 2 were the same.

Day 2, Round 1: Sunday, June 29, 2014, 04:00–09:00 UTC/GMT

Day 2, Round 2: Sunday, June 29, 2014, 10:00–15:00 UTC/GMT

The tasks for Round 1 and Round 2 were the same.

JOI Open Contest, 2015

Round 1: Sunday, June 14, 2015, 04:00–09:00 UTC/GMT

Round 2: Sunday, June 14, 2015, 10:00–15:00 UTC/GMT

The tasks for Round 1 and Round 2 were the same.

JOI Open Contest, 2016

Round 1: Sunday, June 19, 2016, 04:00–09:00 UTC/GMT

Round 2: Sunday, June 19, 2016, 10:00–15:00 UTC/GMT

The tasks for Round 1 and Round 2 were the same.

JOI Open Contest, 2017

Round 1: Sunday, July 2, 2017, 04:00–09:00 UTC/GMT

Round 2: Sunday, July 2, 2017, 10:00–15:00 UTC/GMT

The tasks for Round 1 and Round 2 were the same.

JOI Open Contest, 2018

Round 1: Sunday, July 8, 2018, 04:15–09:15 UTC/GMT

Round 2: Sunday, July 8, 2018, 10:00–15:00 UTC/GMT

The tasks for Round 1 and Round 2 were the same.

JOI Open Contest, 2019

Round 1: Sunday, July 14, 2019, 04:00–09:00 UTC/GMT

Round 2: Sunday, July 14, 2019, 10:00–15:00 UTC/GMT

Round 3: Sunday, July 14, 2019, 15:30–20:30 UTC/GMT

The tasks for Rounds 1, 2, and 3 were the same.

JOI Open Contest, 2020

Sunday, September 6, 2020, 04:00 – Monday, September 7, 2020, 04:00 UTC/GMT

JOI Open Contest, 2021

Sunday, June 6, 2021, 04:00 – Monday, June 7, 2021, 04:00 UTC/GMT

JOI Open Contest, 2022

Sunday, July 3, 2022, 04:00 – Monday, July 4, 2022, 04:00 UTC/GMT

JOI Open Contest, 2023

Saturday, August 5, 2023, 04:00 – Sunday, August 6, 2023, 04:00 UTC/GMT

JOI Open Contest, 2024

Sunday, June 16, 2024, 04:00 – Monday, June 17, 2024, 04:00 UTC/GMT

JOI Open Contest, 2025

Sunday, June 15, 2025, 04:00 – Monday, June 16, 2025, 04:00 UTC/GMT

Since one of the main purposes of the JOI Open Contest was to provide training opportunities for contestants preparing for the IOI, it was usually held about one month before the IOI. The JOI Open Contest was held twice, in 2014 only. In all other years, it was held once.

Until 2019, the contest window was fixed at five hours. Because participants were in various time zones, we held multiple rounds, which imposed a considerable operational burden. Starting in 2020, we introduced a system that allowed participants to choose their own starting time within a 24-hour contest window. This reduced the burden of contest operations.

4. Tasks

In an Open Contest, as in IOI, three tasks have usually been given. The only exception was in 2018, when four tasks were given.

Figure 1 shows the full-score rates and task types of past JOI Open Contest tasks. Based on our experience, the full-score rate of JOI Open Contest tasks appears comparable to, or slightly lower than, what one might expect if the same tasks were used in IOI. Difficulty has varied from year to year. Around 2019–2021, many tasks were relatively easy, whereas since 2022, the task sets have tended to be more difficult, closer to the level of recent IOI tasks.

	2012	2013	2014A	2014B	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025
A	10.9%	<i>0.0%</i>	<u>18.3%</u>	<i>0.0%</i>	???	2.8%	<u>28.3%</u>	16.5%	4.6%	<u>53.9%</u>	<u>50.9%</u>	24.3%	<u>0.6%</u>	23.4%	<u>26.3%</u>
B	<u>57.8%</u>	<i>1.8%</i>	38.6%	43.1%	???	38.7%	11.7%	<i>3.7%</i>	<u>56.6%</u>	<u>33.3%</u>	<u>47.4%</u>	<i>3.7%</i>	<i>0.0%</i>	<i>0.0%</i>	15.1%
C	<i>1.6%</i>	80.7%	<i>0.0%</i>	<u>44.4%</u>	???	6.3%	<i>1.7%</i>	4.6%	<i>1.3%</i>	<u>78.0%</u>	<u>7.5%</u>	11.8%	19.8%	<u>22.8%</u>	<u>1.3%</u>
D								<u>77.1%</u>							

Figure 1. Full-score rates and task types of tasks used in JOI Open Contests. Unmarked tasks are Batch Tasks, underlined tasks are Communication Tasks, and italicized tasks are Output-only Tasks. Data are unavailable only for 2015.

Here are three tasks from previous JOI Open Contests.

JOI Open Contest 2018: “Xylophone”

Problem: There is a hidden permutation $(A[1], \dots, A[N])$ of the integers from 1 to N . A contestant may ask, at most $2N$ times, for the difference between the maximum and the minimum of $A[l], \dots, A[r]$, by specifying l and r . The goal is to determine the entire permutation A . It is guaranteed that the position of N is to the right of the position of 1.

The task can be solved by querying all pairs (l, r) with $r - l = 1$ or 2 . As background, in JOI, Communication Tasks were used only at the Spring Training Camp. Because this task was easier than the Spring Training Camp level, it gave contestants across a wide range of levels an opportunity to experience a Communication Task.

JOI Open Contest 2017: “Bulldozer”

Problem: There are N points in the plane. Point i has coordinates $(X[i], Y[i])$, and its score is $W[i]$. When two parallel lines are drawn, what is the maximum possible sum of the scores of points lying between the two lines? Here $N \leq 2000$.

This task can be solved by considering $O(N^2)$ possible directions of the lines and processing them in angular order. The key observation is that when the direction changes to the next relevant direction, the order of the points in that direction changes only by swapping two adjacent points. This technique for solving such geometric problems is now called the “Bulldozer Trick” in Japan, Korea, and elsewhere.

JOI Open Contest 2023: “Cell Automaton”

Problem: There is an infinite grid. Initially, N cells $(X[1], Y[1]), \dots, (X[N], Y[N])$ are coloured black. Every second, each cell’s color changes as follows: a black cell becomes grey; a grey cell becomes white; and a white cell becomes black if at least one of its edge-adjacent neighbors is black. For each time $T[1], \dots, T[Q]$, determine the number of black cells. Here, $N \leq 100000$ and $Q \leq 500000$.

This task is considered one of the most difficult in the history of JOI Open Contests. During the contest, the highest score was 49 points. It is very difficult to derive an efficient method for computing the intersections of squares centered at the relevant cells. Even after this observation, the implementation remains highly challenging.

5. Technical Aspects

JOI uses the CMS (Contest Management System) to provide a contest environment similar to IOI (Maggiolo & Mascellani, 2012). Rather than using the latest upstream version directly, we have continuously maintained and operated a private repository based on the 1.4 series of CMS, adding our own extensions. JOI Open Contest also uses this infrastructure.

In this section, we describe three technical aspects of our system: maintaining compatibility with older JOI task formats, Auto Scaling and monitoring on AWS (Amazon Web Services), and configuration management as Infrastructure as Code (IaC) to make infrastructure construction reproducible.

First, from an early stage, JOI used a judging system called ImoJudge, which was developed in Japan and widely used domestically (Imajo, 2011). Even now, we maintain compatibility with traditional task packages and task types originating from that system. To this end, we have extended CMS loaders for items such as Contests, Tasks, and Users, so that these older formats can be handled directly within CMS.

Second, we describe extensions to CMS that support operation on AWS. Two major operational issues are reducing server costs and making monitoring efficient. For judging workers, we use c4.2xlarge instances. Among high-performance instances, these are relatively inexpensive, provide high processing speed, and offer stable execution times because memory and cache access are less affected by other instances.

However, keeping a large number of such instances running throughout the contest period would increase costs. We therefore extended CMS to allow the number of workers to be dynamically increased or decreased while the system was running. This made it easier to manually adjust the number of workers based on load and to introduce Auto Scaling. It also enabled the safe use of inexpensive Spot Instances despite their risk of unexpected termination.

As a concrete implementation of this automation, we send the estimated wait time for the judging queue to CloudWatch and use it as an indicator for adjusting the number of instances. Unlike official JOI contests, which end relatively quickly, JOI Open Contest requires long-duration operation. Therefore, cost optimization through dynamic scaling is particularly effective.

For monitoring, we have also built a system that automatically sends notifications about the status of multiple servers and key logs to an internal communication tool. This enables a small team to efficiently monitor the entire infrastructure.

Third, we have developed IaC (Infrastructure as Code) to ensure reproducibility in infrastructure construction and configuration deployment. At JOI, we have built a mechanism that expands Jinja2 templates with configuration inputs and automatically generates Terraform configuration files that define the AWS infrastructure, as well as Ansible inventories that define the destinations for CMS and middleware configuration. This enables the management and deployment of multiple infrastructures in parallel, such as internal testing environments, domestic contest environments, and Open Contest environments, each with slightly different requirements, without duplicating or modifying the core management code.

In this way, JOI Open Contest is operated by combining CMS extensions for compatibility with existing task formats and extensions for utilizing cloud infrastructure. However, due to continuous independent extensions, the codebase has increasingly diverged from the upstream CMS. Keeping up with the latest upstream version and integrating our extensions with it remain technical challenges for the sustainable operation of the system in the future.

6. Conclusion

When JCIOI began organizing Open Contests in 2013, the online competitive programming environment was not as well developed as it is today. CMS, which is used in our Open Contests, was originally used as the contest system for IOI 2012 in Italy. At that time, practicing with IOI-level tasks was difficult, which posed a challenge for contestant training. The same situation likely existed not only in Japan but also in other countries.

When we prepared English task statements and held an Open Contest open to all, we were surprised to see more participants from outside Japan than we had expected. According to the post-contest questionnaire, several contestants who represented their countries at the IOI in the same year also participated in Open Contests. Some contestants participated in Open Contests multiple times. We believe that our Open Contests have contributed to the competitive programming and informatics education communities.

More than ten years have passed since the start of Open Contests. The competitive programming landscape has changed significantly. Many task sets, including IOI-level problems, are now available online, making it much easier than before to hold virtual training contests. On the other hand, preparing original tasks and holding the JOI Open Contest every year have required considerable effort. Taking these circumstances into account, we decided to end the JOI Open Contest (Open Contests of Type 1) using original tasks in 2025.

However, we will continue Open Contests of Type 2 (mirror contests of official JOI contests). We warmly invite everyone interested in JOI tasks and contest environments to participate in these Open Contests.

References

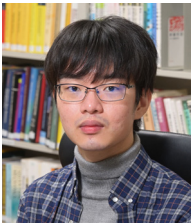
- Imajo, K. (2011). Contest environment using wireless networks: A case study from Japan. *Olympiads in Informatics*, 5, 26–31. <https://ioinformatics.org/journal/INFOL077.pdf>
- Maggiolo, S., & Mascellani, G. (2012). Introducing CMS: A contest management system. *Olympiads in Informatics*, 6, 86–99. <https://ioinformatics.org/journal/INFOL107.pdf>



Tetsushi Ito is an associate professor at Kyoto University. As a high school student, he participated in IOI 1994 and IOI 1995. He has also served several times as a leader or deputy leader of the Japanese delegation to the IOI. He served as Chairperson of the Host Scientific Committee for IOI 2018. He is currently a board member of the Japanese Committee for the International Olympiad in Informatics (JCI-OI) and Chairperson of its Scientific Committee.



Seiji Hirata is an undergraduate student at the University of Tokyo. He is currently a member of the Technical Committee of JCIOI.



Kengo Shimizu is a master's student at the University of Tokyo. From 2022 to 2026, he worked on automating server and contest environment configuration as a member of the Technical Committee of JCI-OI.



Hirotaka Yoneda is a master's student at the University of Tokyo. As a high school student, he participated in IOI 2018 and IOI 2020. Since then, he has competed in the International Collegiate Programming Contest (ICPC), including a second-place finish at the ICPC 2025 World Finals. He has been involved in training JOI contestants and members of the Japanese delegation to the IOI for several years.

Some Puzzles for Brainstorming

Mohammad KAYKOBAD

Department of Computer Science and Engineering BRAC University, Kha 224 Pragati Sarani, Merul Badda, Dhaka 1212, Bangladesh, Fellow, Bangladesh Academy of Sciences
e-mail: kaykobad@bracu.ac.bd

Abstract: For superior performance in IOI or IMO, it is important that students develop their logic and analytical skills. The advent of powerful LLMs is creating an opportunity for students to avoid spending time in brainstorming over problems. This paper provides students with an opportunity to sharpen their analytical skills and logic by solving problems, most of which are drawn from existing resources and are often modified/generalized to challenge their presence of mind. References to some related topics have also been included. At the end of the paper, some hints for the solution have been provided.

Keywords: Logic problems, problems for analytical skill development.

There has been a lot of literature (1, 3-8) published and in online repositories (9-11) that challenge young people with puzzles and mathematical problems for solution. This article compiles some of the interesting puzzles, especially for our young people, to brainstorm over. Some of these puzzles are available in published materials and on different websites. These puzzles are either logic-based or require some analytical skill for their solution. These problems have been listed in random order to test real examinations/tests. The author posed these problems to pre-university students, who found them challenging and interesting.

1. Four students were standing face to face, and a teacher had many red and white caps. The teacher put one cap on each one's head, and we now have only four caps. The teacher asked for raising hands if anyone had seen two red caps. Immediately, all of them raised their hands. Then he said if someone can figure out the color of her own hat, she should put down her hands and tell the color. Nobody was putting down hands for a long time. Then at long last Alice put down her hands and told the right color. Tell the color and present logic behind your answer.
2. A couple invited 6 couples to a dinner party. Assume 'knowing' and 'not knowing' are symmetric relations. Those who did not know each other got to know each other and shook hands. You can assume that husband and wife know each other. After dinner, at the time of departure, the host husband asked everyone else how many people they had shaken hands with. Everybody answered with a different number. How many people did the host husband shake hands with?

3. Somebody starting at point A on Earth's surface walked to the north for 20 kms to reach point B, and then walked 20 kms to the south to reach point C. Is it possible that the distance between A and C is 30 kms?
4. There are 50 coins, sides not recognizable by touch, on the table, of which 16 coins are heads up. You are blindfolded. You can flip as many coins as you wish. Then you should divide them into two piles, not necessarily of equal size, so that each pile has the same number of heads up.
5. Adam and Bill met after a long time to learn that both of them became mathematicians and married. Adam asked Bill how many children he had, to which Bill responded, 'three'. When asked about their ages, Bill said the sum of their ages is 13, and the product of their ages equals the number of windows in the adjacent building. Then Adam said that he needed a little more information to figure out their individual ages. Then Bill said that the youngest child's birthday can be celebrated only once every four years. What are their ages?
6. How can you divide four similar cakes among 63 students so that the smallest cake piece is as large as possible? What about replacing numbers respectively by (2,15), (3,35), (4,63), (5,99)?
7. A jail super decided to set free as many of the 10 long-serving prisoners as could tell the color of the hat he would be given by the jailer. He asked all of them to come to the playground the next day in the morning. He also said that he would have plenty of white and black hats. He would also place the prisoners in a queue at his discretion. Starting with the last person in the queue, he would ask what color the hat he was wearing on. Those who could tell the right color would be freed. Prisoners went to the cell and chalked out a plan to set the largest number of them free. In telling the color of the hat, they cannot say anything other than color, and no other codes can be transmitted any other way. Find the strategy in which the largest number can reach freedom.
Supplementary question: If there are m known colors and n persons, how many people can get to freedom?
8. You are walking on a road that has been divided into two roads, one of which leads to the post office and the other to the university. You want to go to university, but you do not know which road leads to university. There is a guard standing near the fork who knows which road leads to which destination. It is not known whether the guard is truthful or a liar. The guard can be asked only one question to find the right path. What is the question? If the guard alternates on speaking the truth and telling lies what would be the question?
9. There are 30 marbles on the table. Two players play the game. The first player, in his first turn, cannot remove all the marbles. Any player in his move can take at most double the number of marbles his opponent took in his last turn, and should take at least one. The player emptying the table wins the game. How are you going to play the game as the first player? How about starting with n marbles and removing at most k times as much as the opponent's last turn?

10. Four dogs were standing on the corners of a square field of one square mile. Every dog runs exactly to the dog on the right, covering half the distance with its eyes closed, and then stops. Again, run to the dog on the right and stop after half the distance has been covered. What happened to the dogs at the end? How much distance could each dog cross? If the dogs opened after covering $1/3$, $1/4$, or $1/n$ of the distance, what would the answers be?
11. Two brothers had some goats. Each goat was sold at a price equal to the number of goats they had. They started sharing the money, removing 20 euro bills in turn, with the elder brother taking the first bill. In his turn, after a while, the elder brother saw that if he took another 20-euro bill, there would not be enough to share equally with the younger brother. So he gave his younger brother his knife, which cost an integer Euro. What was the price of the knife?
12. It is said that the following problem was posed to an ailing Einstein in the hospital by his journalist friend with the hope that, by thinking about a solution to the problem, Einstein would forget his pain, and Einstein proved his friend wrong. In how many different places can swapping the hour and minute hands of a clock result in a valid time?
13. A robot is standing by the side of an infinite wall with a single door that can be discovered only when the robot is at step exactly in front of the door. Devise an algorithm for the robot that, in the worst case, discovers the door in the minimum number of steps.
14. Supplementary problem: Assume you are lost in a forest at a junction with 3 emanating roads, one of which has a treasure at an unknown distance, which you can recognize only if you are just in front of it. Design an algorithm that will lead you to the treasure in the minimum number of steps in the worst case.
15. Charlie is a new friend of Alex and Ben. They want to know Charlie's date of birth. Charlie tells the following possible dates: 15 May, 16 May, 19 May, 17 June, 18 June, 14 July, 16 July, 14 August, 15 August, and 18 August. Charlie whispered her month of birth to Alex and the date to Ben. Alex said, "I do not know Charlie's date of birth. But I am sure neither does Ben." Ben said, "At first I did not know Charlie's birthday, but now I know." Alex said, "Then I also know." Find Charlie's date of birth.
16. You are in a dense convex forest of S sq.km area. You have an instrument that will allow you to follow any trajectory you wish. What is the minimum distance in the worst case you need to cross to come out of the dense forest?
17. A hard-working father called his idle son to his deathbed and said he had bought gold with all his earnings and hidden it in the forest. The son wanted to know the location of gold, to which the father said, "There are two identical trees, A and B, in the forest, and a stone S. You must walk from S to A and then walk an equal distance perpendicularly towards the other tree to find point C. You should again walk from S to B, and an equal distance perpendicularly to find D. Gold is at the midpoint of A and B." The son went to the forest and found trees A and B, but failed to find S. Can you help him find gold?
18. A pure, unrealistic mathematical problem. Five friends caught some fish and fell asleep. The first friend woke up at night, divided the fish into five equal shares, and

threw the remaining two into the river. He took his share, gathered the remaining fish, and went to sleep. In this way, each friend did exactly the same as the first friend. In the morning, all five woke up at the same time and found exactly the number of fish they caught. What was the minimum number of fish caught?

19. Find a ten-digit number whose leftmost digit is the number of 0's in it, next digit is the number of 1's, and in this way, the rightmost digit is the number of 9's in it. What is the number?
20. Find a method of representing any integer using three 2's and known functions (no use of +, −, ×, / signs and no other digits).
21. A boy went to buy a pen, a pencil, a notebook, and an eraser. Then the shopkeeper told the amount to pay. The boy asked how the shopkeeper found the total amount, and the shopkeeper answered that he multiplied all the prices. The boy said the shopkeeper was wrong, to which the shopkeeper remarked, even adding the values gave the same answer. What were the prices?
22. If two students are to undertake a journey of d kilometers using a very well-trained horse that can carry one of the friends at a time. Assuming horse speed $v_h \gg v_m$ (speed of man), what will be the minimum time required to complete the journey for the friends to start their journey together and end it in the minimum time? What about if there were n men and m horses and each horse could carry c men at a time?
23. Given two straight lines in 3D space, how do you find their distance? What is the answer if you are given two straight line segments?
24. If you invest 100 euro at a rate of interest euro 100/per year. You can get an interest rate based on the time you invest: 3 months will give you 25 euros, and 1 month, 100/12 euros. You can invest for any small time period you wish and will receive interest proportionate to the period. What is the maximum amount of interest you can earn in a year?
25. Given a directed graph $G = (V, E)$ with nonnegative weights on edges. By spanning tree, we mean the spanning tree of the underlying graph (without directions). Every edge (a, b) of the spanning tree uniquely determines a fundamental cut set and partitions the vertex set V into V_a and V_b . Weight of cut set $[V_a, V_b]$ = weight of edges of the cut set in the orientation of edge (a, b) minus weight in the reverse direction. Is it always possible to have a spanning tree for which the weight of every cut set is non-negative? (Kaykobad, 1986).
26. A pen, an eraser, and a notebook cost US\$100. The price of two pens is less than the price of a notebook. The price of four erasers is less than the price of three pens, and the price of a notebook is less than the price of three erasers. If all the prices are integers, find each price.
27. Is it possible that in a race, a pair of legs of a 27-foot-long, 9-foot-high, and 7-foot-wide elephant ran 27 feet more than the other two legs?
28. In the month of January of a year, there were 4 Sundays and 4 Wednesdays. Which day was the 10th of January?

29. Mr. Johnson is now as old as the ages of his twin sons and three daughters taken together. In year 20xx, he will be as old as the sum of his daughters' ages. What will be the ages of twin sons in year 20xx compared to their current ages?
30. Grandpa and grandma are together 140 now. What is grandma's current age if grandpa's age was twice grandma's when grandpa was as old as grandma now?
31. If the sum of the digits of a pair of numbers is divisible by k , what is their smallest difference?
32. John said, "The day before yesterday I was 10, but I will turn 13 next year." What is the birthday of John?
33. How can I cut the 3×8 cm² board into only two pieces so that they will fill inside the 2×12 cm² hole exactly?
34. A girl, a boy, and a dog are walking down a road together. The boy walks at 5 km/h, and is just behind the girl who walks at 6 km/h. The dog runs from boy to girl and back again with a constant speed of 10 km/h. The dog does not slow down on the turn. How far does the dog travel in 1 hour?
35. If I drive halfway to the city at 40 km/hour, how fast do I have to go for the rest of the way to make the average speed 80 km/hour for the entire journey?
36. If $x^x = a$, then what is the value of x ?
37. Find $\frac{1}{60} = \frac{1}{A} + \frac{1}{B} + \frac{1}{C}$. What about solutions to $\frac{1}{60} = \frac{1}{A} + \frac{1}{B} + \frac{1}{C} + \frac{1}{D}$ or $\frac{1}{60} = \frac{1}{A} + \frac{1}{B} + \frac{1}{C} + \frac{1}{D} + \frac{1}{E}$? All denominators are positive integers.
38. What is the minimum number of comparisons for sorting 5 numbers when the result of each comparison is known immediately, and can be used for the decision for the next comparison?
39. A pen, two erasers, and 3 workbooks cost euro 170, 3 pens, an eraser, and two workbooks cost euro 180, and two pens, 3 erasers, and a workbook cost eur 190. How much does a pen cost?
40. There are 3100 coins laid out in a row. For some positive integer x at least $\frac{3}{4}$ of the first x coins are heads, and at least $\frac{4}{5}$ of the last x coins are tails. What is the maximum possible value of x ?
41. N mixed, and messed-up electrical wires are hanging from a skyscraper. You need to determine their heads and tails by tying some of them in one end and testing connectivity at the other. If there are two, you cannot properly identify the ends. If 3, tie two and name the other as 1. Go to the roof to identify the other end of the untied wire and so on. What is the minimum number of times you need to go up and down to identify heads and tails of all wires?

Hints: 2. Draw a graph in which edges indicate familiarity 4. Figures are important 5. Ties need to be broken, 7. a) the last person should answer depending upon the parity of the color of the white/black hats in front of him. b) You can use the EXOR operation 8. Ask a question to which the guard's answer will be the same whether he speaks the truth or lies. Double-positive or double-negative statements give the same answer. 9. Start developing solutions for smaller figures, 11. Study remainders 12. Figure out the smallest interval after which swapping hands give right time. 13, 14. Analyze complexity based on strategy. 15. Hint: Use the method of elimination. 16. Hint: Follow the boundary covering a given area but of minimum length. 17. Hint: The gold can only be discovered if its position is independent of the position of the stone.

References

- Fedorov, R., Belov, A., Kovaldzhii, A. & Yashchenko, I. (2011). *Moscow Mathematical Olympiads, 1993-1999* (MSRI Mathematical Circles Library).
- Fedorov, R., Belov, A., Kovaldzhii, A. & Yashchenko, I. (2011). *Moscow Mathematical Olympiads, 2000-2005* (MSRI Mathematical Circles Library, 7).
- Gardner, M. (1994). *My Best Mathematical and Logic Puzzles* (Dover Recr Math) (Dover Math Games & Puzzles)
- Kalman, R. (Ed. 2008). *Math Olympiad Contest Problems, Volume 2* (revised edition).
- Kaykobad, M. (1986), Minimum connection time and some related complexity problems, Ph.D. Thesis, The Flinders University of South Australia, Bedford Park, South Australia, Australia.
- Kordemsky, B. A. (1992). *The Moscow Puzzles: 359 Mathematical Recreations* (Dover Math Games & Puzzles).
- Lenchner, G. (1997). *Math Olympiad Contest Problems for Elementary and Middle Schools, Vol. 1*.
- Perelman, Y. (1985). *Mathematics Can Be Fun*, Mir Publishers, Moscow. Winkler, P. *Mathematical Puzzles*. Online (Accessed on 21 Mar. 2026): <https://math.dartmouth.edu/news-resources/electronic/puzzlebook/book/book.pdf>.
- Puzzlewocky: brain teasers, word games, paradoxes, situation puzzles, and optical illusions. Online (Accessed on 21 Mar. 2026): puzzlewocky.com/
- Problems and Solutions in Mathematical Olympiad. Online (Accessed on 21 Mar. 2026): <https://www.scribd.com/document/713191859/Problems-and-Solutions-In-Mathematical-Olympiad-Vol-1-2021>.



Dr. Mohammad Kaykobad is a Distinguished Professor in the CSE Department at BRAC University, Dhaka, Bangladesh. Prior to this, he was a Professor in the CSE Department at Bangladesh University of Engineering and Technology. He received an M.S. (Hons.) in Engineering from OMEI, now Odessa State Maritime University, in 1979, an M.Eng. Degree in 1982 from AIT, Thailand, and a Ph.D. from Flinders University of South Australia in 1988. Dr. Kaykobad is a Fellow of the Bangladesh Academy of Sciences (BAS) and served as an Associate Secretary of its Executive Council.

Dr. Kaykobad was the organizing chair of the first-ever international computer conference, ICCIT, held in Bangladesh. He also served as the chair of the steering committee for the Workshop on Algorithms (WALCOM), with proceedings published in *Lecture Notes in Computer Science* by Springer. He has mentored more than a dozen undergraduate students whose research was published in reputable international journals. He is a frequent columnist in national newspapers, authoring over 300 articles, most of which focus on the country's education. He was a pioneer in introducing Mathematics Olympiad, Informatics Olympiad, and Science Olympiad in Bangladesh. He has led or served as deputy leader of the Bangladesh IOI team since 2008, which has won a gold medal, seven silver medals, and twenty-five bronze medals. Currently, he is the president of the Bangladesh Mathematical Olympiad Committee and the Bangladesh Informatics Olympiad Committee. He was the director of the ICPC Asia Region Dhaka site from 2001 to 2003. He served as the chief judge of the IIT Kanpur Asia Region site of ICPC in 2010, 2011, and 2013. He received the Outstanding Coach Award from ICPC in 2002 in Honolulu, Hawaii, the Senior Coach Award at the ICPC World Finals held in St. Petersburg, Russia, in 2013, and the ICPC Lifetime Coach Award in 2019 in Porto, Portugal. He was also awarded a gold medal by the Bangladesh Physics Olympiad Committee.

Dr. Kaykobad was a Visiting Professor in the CSE Department at the Chinese University of Hong Kong, Kyung Hee University in Korea, ANU, Monash University in Australia, and Amritapuri University in India. In 2005, the country's president awarded him a Gold Medal from the Bangladesh Computer Society in recognition of his contribution to the country's computer programming culture. In 2006, he received the BAS Gold Medal in the physical sciences senior category, which was presented by the country's Prime Minister.

Operational Staffing for an IOI Host: A Practical Role-by-Role Guide

Eljakim SCHRIJVERS

Stichting Nederlandse Informatica Olympiade

e-mail: eljakim@gmail.com

Abstract. Hosting the International Olympiad in Informatics (IOI) requires the coordinated effort of dozens of operational staff and volunteers. This document provides a structured, role-by-role view of that effort: which positions must be filled, when each becomes active, what they are accountable for, and the milestones that keep preparation on track. It is aimed at teams moving from committee formation to operational execution. The recommendations are grounded in my own direct involvement with the IOI community over many years. As IOI Treasurer, I visited successive host countries in the run-up to each IOI and observed their operational preparations from the inside, and I was also the main organiser of EGOI~2024 (European Girls' Olympiad in Informatics) in the Netherlands, where I was responsible for the entire event from start to finish. The material presented here has been discussed with several past IOI hosts, whose feedback has shaped the content.

1. Introduction

The International Olympiad in Informatics is an annual event bringing approximately 700 participants from around 95 delegations to a single location for eight days. A typical edition has roughly 380 contestants (four per delegation), 190 team leaders and deputy leaders (two per delegation), a guide corps with one guide per delegation plus reserve and Superguides, and another 40 or so IOI committee members and host organisers who are present on-site throughout the week. On top of that, the host recruits a substantial pool of additional volunteers for packing, check-in, the airport, the contest hall, the Fun Zone, ceremonies, excursions, and daily transport. Contestants compete in two programming contests while team leaders participate in a sequence of General Assembly meetings that determine tasks, translations, and final results.

While the high-level structure of an IOI (the Host Steering Committee, the Host Organising Committee (HOC), the Host Scientific Committee (HSC), and the Host Technical Committee (HTC)) is documented in the IOI Regulations (IOI, 2021) and has been described in prior host experience reports (Yusubov et al., 2022), and earlier general-aspects guidance for hosts exists in the literature (Matkarimov et al., 2017), the translation of that structure into an operational staffing plan is largely absent. A host team that has formed its committees still faces a large number of practical questions that are not answered by any of

those sources: Who is the single contact point for team leaders in the days before arrival? Who owns the T-shirt sizes? Who posts the departure schedule to leaders, and by when? Who is standing at the airport for an eighteen-hour shift on arrivals day?

This document is intended as a *guideline*, not a prescriptive rulebook. Every IOI host operates in a different country, with different resources, venues, cultural norms, and volunteer pools. The roles, structures, and recommendations described here reflect what has worked well in the past, but they do not need to be followed to the letter. Hosts are encouraged to adapt, merge, or reorganise roles to fit their own circumstances. What matters is that every responsibility described here is covered by someone, not that the org chart matches this document exactly.

The focus throughout is strictly *operational*: how the host staffs and runs the week around the competition. The competition itself (contest rules, task specification, grading, and the internal working methods of the HSC and HTC) is out of scope; for those, readers should consult the IOI Regulations (IOI, 2021).

The rest of this document is a practical staffing blueprint: a map of the roles a host needs to fill, the responsibilities each one carries, and the milestones by which each role becomes active. It is organised as follows. Section 2 describes the rhythm of the IOI week. Section 3 draws the distinction between governance and operations. Section 4 presents the full operational org chart. Section 5 sets out the HOC Chair's ongoing responsibilities across preparation and the event week. Sections 6–14 then catalogue each operational domain and the roles within it, from Participant Care through to Accommodation. Section 15 covers the post-event phase, and Section 16 closes with reflections on scope, what hosts tend to underestimate, and what could usefully be shared across future editions.

2. Structure of an IOI Week

For operational planning, the rhythm of a typical IOI week is given in Table 1. The event spans eight days.

Table 1: Typical IOI week programme.

Day	Programme
Day 1	Arrivals. Check-in, goodie bag distribution, welcome activities.
Day 2	GA 1 (pre-practice). Practice session. Opening ceremony. GA 2 (post-practice). GA 3 / first translation night.
Day 3	Competition Day 1 (5 h). IOI Conference. Analysis and appeals. IOI Fun Zone. GA 4 (post-contest).
Day 4	Small excursion. GA 5 / second translation night.
Day 5	Competition Day 2 (5 h). Leader Work Groups. Analysis and appeals. IOI Fun Zone. GA 6 (post-contest).
Day 6	Big excursion.
Day 7	Tech talks / Knowledge Fair. Final GA 7 meeting. Closing ceremony and awards banquet.
Day 8	Departures.

Throughout the week, team leaders attend General Assembly meetings while contestant activities run in parallel. Table 2 lists the seven GA meetings, their placement in the programme, and the purpose of each meeting.

Because contestant and leader activities run in parallel on most days, the host team must operate several programme tracks simultaneously. Understanding this concurrency is essential when assigning operational roles: most staff will need to be in only one place at a time, so roles must be designed around that constraint.

Table 2: General Assembly meetings during the IOI week.

Meeting	Details
GA 1	Day 2 morning, before the practice session and opening ceremony <i>Welcome; discuss practice session and competition rules.</i>
GA 2	Day 2 afternoon, after the practice session, before quarantine starts <i>Results from practice session; general announcements. Leaders need time with contestants afterwards to relay outcomes.</i>
GA 3	Day 2 evening, at the start of the first translation night <i>Selection and Translation of Competition Day 1 tasks.</i>
GA 4	Day 3 afternoon, after Competition Day 1 <i>Appeals from Competition Day 1.</i>
GA 5	Day 4 evening, at the start of the second translation night <i>Selection and Translation of Competition Day 2 tasks.</i>
GA 6	Day 5 afternoon, after Competition Day 2 <i>Appeals from Competition Day 2; presentation of candidates for elections.</i>
GA 7	Day 7 morning, final GA meeting, before the closing ceremony <i>Confirmation of results; elections and votes.</i>

2.1 Quarantine

There are two separate quarantine periods, one per competition day. Quarantine begins *before* the contest tasks are handed out to the leaders: contestants and leaders must already be physically separated by the time the GA session starts. The first quarantine runs from the evening before Competition Day 1 (first translation night) until the end of that contest; the second runs from the evening before Competition Day 2 (second translation night) until the end of that contest. Between and outside these periods, contestants and leaders mix freely.

Quarantine affects accommodation, meals, transport, and guide assignments. In the current digital age, full quarantine is practically impossible, but the host must avoid situations where contestants and leaders who know the tasks come into direct contact. Quarantine serves two purposes: preventing leaks and preventing even the appearance of leakage. If an unexpected result occurs, other delegations may recall seeing a leader and contestant

together on the morning of the contest, putting the host, the GA, and the delegation concerned in a very difficult position.

Each quarantine begins at the start of the corresponding translation night, i.e. at the start of GA 3 for the first contest day and at the start of GA 5 for the second.

Equally important is ensuring sufficient contact time *before* quarantine begins. After GA 2 (which covers the results of the practice session), leaders must have time to meet with their contestants and relay anything important that came out of the discussion. The schedule should therefore guarantee a window between the end of GA 2 and the start of quarantine in which leaders and contestants can freely meet.

3. Governance vs Operations

Hosts frequently conflate two distinct layers of organisation that must work in parallel but serve different purposes: the *governance* layer and the *operational* layer. Getting this separation right early avoids structural confusion later.

3.1 Governance: the Host Steering Committee

The Host Steering Committee is a small, high-level body whose function is political rather than operational. In countries where the government is directly involved in hosting the IOI, its members typically include a government minister-level Chair (from the Ministry of Education or the Ministry responsible for digital affairs), co-chairs from other relevant ministries (Foreign Affairs, Internal Affairs, Transport), and the individual who originally championed the IOI bid.

However, government involvement is not universal: in some countries the IOI is hosted by a foundation, university, or industry consortium with little or no direct government participation. In those cases the Steering Committee may consist of senior representatives from the hosting institution and key sponsors, fulfilling the same function of unlocking resources and resolving high-level blockers.

The Host Steering Committee's role is to unlock resources that the operational team cannot access directly: national budget allocations confirmed by government letter, visa facilitation requiring Foreign Ministry involvement, security clearances, access to state-owned venues, and rapid resolution of administrative or political blockers. It does not make operational decisions. It does not set the bus schedule, select the T-shirt design, or determine the contest network topology.

3.2 Operations: the Host Organising Committee

The HOC is the working body, led by the HOC Chair, who holds national-level authority and is the primary liaison to the International Committee. In theory, the Chair's function is not day-to-day task execution but rather holding the overall schedule, budget, and risk

register, ensuring crossdomain coordination, escalating to the Host Steering Committee when blockers require political or financial intervention, and representing the host to the IC. Liaison with the ISC and ITC is handled by the HSC and HTC chairs respectively; the HOC Chair receives reports from the HSC and HTC but does not sit on the ISC or ITC. The day-to-day and week-of-event responsibilities of the Chair are covered in detail in Section 5.

In practice, however, the HOC Chair quite often takes on one or more of the functional coordinator positions described in this document, for example doubling as the Venues Coordinator or directly managing registration. When this happens, the Chair role becomes a de facto day-to-day operational position as well, with the coordination responsibilities on top. This is workable as long as the Chair remains aware of the dual role and does not lose sight of cross-domain coordination while handling their own operational tasks.

Regardless of whether the Chair takes on functional roles, one aspect of the position is paramount: they must have a clear mental picture of how to run the entire IOI, covering all the operational aspects described in this document. Because the IOI rotates each year between countries that rarely host it more than once, almost every host team is effectively operating for the first time, and the Chair is the one member of that team who cannot afford to be. The HOC Chair should therefore have first-hand experience from multiple previous IOIs, whether as a team leader, committee member, or in another operational role. One edition seen from the inside is rarely enough to hold the complete picture; a Chair who has been through several knows what recurs, what changes between hosts, and what the IC and the committees expect. The individual coordinators typically do not have deep knowledge of the IOI as a whole; they know their own domain. It is the Chair who holds the complete picture and who understands how all the pieces fit together. This means they should hold preparatory meetings with the core team well in advance, not only to discuss each coordinator's own responsibilities but to give everyone a general understanding of the full event. When every coordinator understands the whole show, not just their part of it, cross-domain problems are spotted earlier and coordination during the event becomes far smoother.

The rule of thumb is: if a decision requires spending money, invoking governmental authority, or engaging an external institution, it goes to the Host Steering Committee. If a decision concerns how the event runs (schedules, assignments, room layouts, volunteer briefings) it stays within the HOC. The HOC Chair is the routing point between the two layers.

3.3 The scientific and technical layer

The HSC and HTC sit alongside the HOC and report to the HOC Chair, but are governed by the international committees (ISC and ITC respectively). The HOC interacts with them at defined operational interfaces: task translation logistics (Translation Night Coordinator) and contest hall infrastructure (Contest Hall Coordinator). These are described in Section 9.

The HSC in particular operates almost entirely outside the operational staffing plan: its

work (selecting and preparing tasks, producing test data, building graders, coordinating with the ISC) is scientific rather than operational, and is largely invisible to the rest of the host team. For that reason the HSC is not covered in depth in this document beyond the interfaces where it meets operations. The HTC sits closer to the operational side because the infrastructure it runs (network, power, workstations, CMS, printing, the translation system) has to be planned and installed inside the host's venues; where the HTC shows up in this document, it is through those infrastructure interfaces rather than as a committee in its own right. Readers looking for the internal structure and working methods of the HSC and HTC should consult the IOI Regulations (IOI, 2021).

3.4 *Adjacent IOI-wide and host functions*

Several roles are referenced throughout this document but sit *outside* the host's operational structure. They belong either to the IOI internationally or to the host's non-operational functions, and the coordinators described in the rest of this document interact with them at specific points rather than having them in their reporting line.

IOI President. The President of the IOI leads the IOI internationally, chairs the International Committee, and represents the IOI as its public figurehead. They do not sit inside the HOC, but are a central figure at the IOI week: it is customary for them to deliver opening remarks at both the Opening and Closing Ceremonies, and to oversee the IOI Flag Handover to the following year's host.

IOI Treasurer. The Treasurer looks after the long-term, IOI-wide sponsor relationships that persist across editions, and owns those sponsor contracts. On behalf of the IOI, they negotiate multi-year commitments with major sponsors and ensure that their contractual visibility requirements are met at each edition. For the host, they are the reference point on anything concerning these long-term sponsors: what has been promised, what can be asked of them, and what is offlimits. The Fun Zone Coordinator, the Branding Coordinator, and the Press & Media Lead all have working contact with them.

IOI Secretary. The Secretary handles formal IOI business that the host should not decide unilaterally, most notably the admission of new participating countries and observer countries into the IOI. Requests of that nature are routed to them, not handled independently by the Registration Lead.

Host sponsorship team. In parallel with the operational staff described in this document, the host typically runs a separate sponsorship function whose job is to recruit and manage *host-specific* sponsors for the current edition (as distinct from the long-term IOI-wide sponsors, which the IOI Treasurer handles). This team is not part of the operational staffing plan, but feeds evolving sponsor requirements into the operational side as activations are refined. The Branding Coordinator works closely with their counterparts in this team to translate sponsor commitments into visible branding at the venues, and the Press & Media Lead coordinates with them on social media activations and external communication. Depending on the host's setup, this function may sit under the Host Steering Committee,

under a separate fundraising body, or inside the HOC as a non-operational role reporting directly to the HOC Chair.

4. Operational Leadership and Workforce

Operational execution is the responsibility of the Host Organising Committee. Table 3 lists the operational domains that report directly to the HOC Chair; each domain's internal roles are detailed in the corresponding section. The operational interfaces with the HSC and HTC fall under Competition Operations (Section 9); the committees themselves are governed by the International Scientific Committee (ISC) and the International Technical Committee (ITC) respectively, both of which sit under the International Committee (IC) of the IOI.

Table 3: Operational domains reporting to the HOC Chair.

Domain	Scope
Participant Care	Guide corps, medical support
Arrivals & Departures	Arrival/departure schedules, airport operations, check-in venue
Venues & Facilities	Contest hall, GA room, Fun Zone, branding, catering
Competition Operations	CMS, translation nights
Communications & Registration	Participant registration, communication channels, press, newsletter (optional)
Ceremonies	Opening and closing ceremonies
Excursions	Day trips and associated logistics
Transportation	Daily bus coordination between all venues
Accommodation	Hotel contracts and room assignments

This document describes over twenty distinct coordinator roles. That does not mean twenty people are needed. In practice, the main coordinator functions at a typical IOI are handled by a core team of around five to eight persons, each covering multiple roles. The exact distribution depends on the host's team and circumstances, but 4 on the following page shows one realistic way of combining roles.

This is only a suggestion. Some hosts will split roles differently depending on the strengths of their team, and some roles (notably Head of Guides) are demanding enough to occupy one person full-time. The important thing is that every responsibility listed in this document is explicitly assigned to someone. The umbrella Venues & Facilities Coordinator (Section 8) is not a separate row in Table 4: that role is the lead of the Venues bucket, and on a small core team it is typically absorbed by the HOC Chair.

Table 4: Example of how coordinator roles can be combined across a small core team.

Combined role	Coordinator roles covered
Participant Care	Head of Guides, Medical Coordinator
Arrivals & Transport	Arrivals Coordinator, Departure Coordinator, Airport Coordinator, Transportation Coordinator, Check-in Venue Coordinator
Venues	Contest Hall Coordinator, GA Room Coordinator
Activities	Fun Zone Coordinator, Excursions Coordinator
Hospitality	Catering Coordinator, Accommodation Coordinator, Ceremonies Coordinator
Competition Operations	CMS Lead, Translation Night Coordinator, Contest Entrance Coordinator
Communications	Registration Lead, Internal Communications Coordinator, Press & Media Lead, Branding Coordinator

4.1 Guides

Guides are the most visible operational face of the IOI for delegations. Each team is assigned a dedicated guide who accompanies them throughout the event: from airport arrival, through meals, transfers, ceremonies, excursions, and contest days, to final departure. Guides are with the contestants essentially all of their waking hours, and for most participants the guide is the person they interact with most, and the most reliable channel through which the host communicates practical information to the team. When things go smoothly for a delegation, it is usually because the guide is on top of the schedule and knows where to find answers; when things go wrong, the guide is usually the first to notice.

The guide corps is recruited, trained, and managed by the Head of Guides. The role is covered in detail in Section 6.1; here it is enough to note that the guide corps is one of the largest pillars of the operational structure, and that it sits alongside the wider volunteer pool described next.

4.2 Volunteer management

The guide corps is only one part of the host's volunteer pool. Packing, check-in, the airport, the contest hall, the Fun Zone, ceremonies, excursions, translation nights, and daily transport together need far more hands than the guide corps alone can supply, so hosts invariably recruit an additional pool of volunteers, typically students or young professionals drawn from local universities, schools, or community organisations. Guides often help out with some of these auxiliary tasks outside their direct delegation duties (welcoming teams at the airport, packing bags in the days before arrivals, supporting the contest hall entrance check, or staffing the Fun Zone), and this is natural and should be encouraged, but not every task can or should be given to guides.

Recruiting, briefing, scheduling, and caring for this wider volunteer base is a cross-cutting responsibility that should not be left to each coordinator individually. In practice, it is helpful to have one person on the core team who owns the overall volunteer plan: a single roster, a single onboarding process, a single point of contact for volunteers' logistical and administrative questions (where to go, what to wear, when meals are served, who to call when shifts go wrong), and a single point of escalation when something breaks down. For the content of any specific task, that same volunteer then reports to the relevant topical coordinator: packing volunteers work under the Check-in Venue Coordinator, airport volunteers under the Airport Coordinator, ceremony ushers under the Ceremonies Coordinator, and so on. The distinction is between *logistical ownership of the volunteer* (one central person) and *task-level direction on the day* (the relevant coordinator). This person may be the Head of Guides acting in an extended capacity, or a separate volunteer lead; what matters is that the responsibility is explicitly assigned rather than diffused.

4.3 Hotels and venues: decisions that sit ahead of this plan

Two decisions sit ahead of everything else in this document: the **hotel inventory** for all participants and the **competition and ceremony venues**. The ceremony venue alone needs a minimum of 1 000 seats (see Section 11), which is one of the largest physical constraints on the search. These go hand in hand and should be reserved together, because each constrains the other in dates and in location. Both should be locked within the **24 to 12 months before the IOI** window: starting around 24 months out so that popular hotels and venues can still be held, and completing by about 12 months out so that the host, the location, and the venue can all be confidently announced at the previous IOI, and so that subsequent operational work (network specifications to the ITC, A/V contracting, catering tenders) has a stable base to build on. The role cards in later sections quote *Active-from* dates for day-to-day operational work, not for the moment the underlying venue or hotel was first locked. Until the relevant venue coordinators come on board, these reservations are owned by the HOC Chair.

4.4 Preparation timeline at a glance

Table 5 gives a single temporal view of the preparation timeline, pulling together the *Active-from* dates and key milestones scattered across the role cards in the rest of this document. The rows below each represent a band of activity rather than a precise deadline; the detailed milestones for each role sit in the relevant section.

5. The HOC Chair's Ongoing Responsibilities

The HOC Chair carries a set of responsibilities that run throughout the preparation period and into the event week itself. Some of them (daily coordinator meetings, the incident escalation path, acting as final decision-maker) only take hold when the week begins; others (the risk register, contingency planning, operational budget ownership) run for the full

preparation period and are referenced throughout this document. The six practices below cover both and should be in place before Day 1.

Daily coordinator meeting. The core coordinators meet once a day, typically in the morning before the programme starts, for a 15–20 minute standing briefing: status of the previous day, known risks for the day ahead, and any cross-coordinator hand-offs. This meeting is the single forum where the full picture is visible at the same time.

Incident escalation path. When something goes wrong on the ground (a delegation lost in transit, a medical incident, a technical failure), everyone involved needs to know who to call. The default is to escalate within the affected domain first (volunteer → coordinator → domain lead), and only cross to the HOC Chair when the issue spans multiple domains or needs immediate authority to commit resources. Medical emergencies are the exception and follow the direct-to-medical-staff protocol described in Section 6.2.

HOC Chair as final decision-maker. During the week the HOC Chair is the final operational decision-maker. When a call needs to be made quickly and coordinators disagree, the Chair decides. Making this explicit in advance, as a normal feature of how the week runs, removes friction in the moment and keeps disagreements from turning into delays.

Table 5: Host preparation timeline at a glance.

When	Main activity
24–12 months out	Hotel inventory and competition/ceremony venues locked (see above). Accommodation Coordinator active.
10+ months out	Venues & Facilities Coordinator and Contest Hall Coordinator active; contest-hall network and power specifications submitted to ITC.
6 months out	CMS Lead active (test installation operational). Ceremonies Coordinator active; ceremony A/V team contracted. Host website carries schedule, visa guidance, and competition rules.
5 months out	Workstation procurement finalised. Fun Zone Coordinator active; Fun Zone layout confirmed and communicated to sponsors.
4 months out	Registration Lead active; registration opens; invitations sent to all countries.
3 months out	Head of Guides, Arrivals, Airport, Branding, Catering, Excursions and Press & Media roles become active. Airport-authority contact begins. Pickup points published on website. First registration chase.
2 months out	Medical, GA Room, Internal Communications, and Transportation roles all active. Visa support processed.
6 weeks out	Data locks: dietary counts and T-shirt sizes exported from registration. First guide briefing. Sponsor items delivery confirmed. GA-room Wi-Fi load test.
1 month out	Translation Night Coordinator and Departure Coordinator active.
2–3 weeks out	Airport site visits. Second guide briefing. Contest Entrance Coordinator active; contest-entrance prohibited-items list confirmed.
Day –3 to –1	Packing, full venue setup, dress rehearsals, medical rooms set up, guide kit issued, final briefings.
Days 1–8	Execution.

Contingency planning. Redundancy at the staffing level is necessary but not sufficient. A small number of event-level scenarios should be planned for explicitly before Day 1: the contest hall becoming unusable (power failure, water damage, security incident), a sustained network or CMS outage during a contest, an illness outbreak large enough to affect the guide corps or the contestant group, a venue evacuation during a ceremony, and the loss of a key coordinator during the week. For each, the HOC Chair should know in advance who makes the call, who executes the fallback, and how the decision is communicated to the delegations. The rules governing the contest itself (postponement, partial-day scoring) are set by the ISC and ITC and sit outside the host's scope; the host's responsibility is the operational fallback around them: an alternative hall, an evacuation route, a stand-in coordinator, an overnight replanning window. A short written scenario list, agreed with the HSC and HTC chairs, is enough; the aim is not an exhaustive playbook but that no one is inventing the response on the day.

Risk register. The HOC Chair maintains a living risk register throughout the preparation period: a short document listing the risks that could derail the event, their likely impact, the mitigation in place, and the owner of each mitigation. The register is reviewed at a recurring HOC meeting, not filed and forgotten. It is tempting to treat the register as a formality; in practice it is the single artefact that forces cross-domain risks to the surface, because individual coordinators tend to see only the risks inside their own domain. Keep it short (one page is usually enough) and keep it current.

Operational budget ownership. Separately from the governance budget held at the Host Steering Committee level, the HOC runs an operational budget covering catering, transport, swag, printing, signage, and the many smaller line items that accumulate during the week. One person on the core team must own this budget: tracking commitments, approving variances during the event when a coordinator needs to spend beyond plan, and reconciling invoices after the event. In small host teams this is the HOC Chair; in larger ones it is a dedicated finance lead reporting to the Chair. Either way, the role must be named before the preparation phase begins. Financial close-out runs on for months after the event, long after the rest of the HOC has stood down, and is itself a reason to appoint a dedicated finance person even when the preparation budget looks small enough to fold into the Chair's role.

6. Participant Care

Participant Care covers the welfare and day-to-day accompaniment of all participants, with particular attention to contestants who are usually minors. The guide corps, its management structure, and the medical function all sit within this domain.



Figure 1: Roles within Participant Care.

6.1 Head of Guides

Reports to: HOC Chair

Headcount: 1 Head + 1 Deputy (Communications) + Superguides + Guides + reserve pool

Structure: One Superguide per 8–10 guides; reserve pool of 5–8 spare guides

Active from: 3 months before IOI

The Head of Guides recruits, trains, and manages the entire guide corps (see Section 4.1). Although the guides themselves are the visible face of the host for delegations, the Head of Guides works mostly behind the scenes: building the roster, managing the reserve pool, running briefings, and handling escalations that Superguides cannot resolve on their own. Other olympiads, such as the EGOI (European Girls' Olympiad in Informatics), have operated successfully with one guide per two or three delegations, but one guide per delegation remains the norm for the IOI given its scale.

The **Deputy for Communications** has one well-defined function: all written communication to and from guides flows through this person. This includes the group messaging channel, daily shift reminders, briefing distribution, and incident escalation. Separating this from the Head of Guides ensures that the Head can focus on operational and welfare decisions without becoming a communication bottleneck.

The **Superguide layer** provides the first tier of escalation for problems a guide cannot resolve alone, like a missing contestant or a welfare concern.

The **reserve pool** is rostered daily; when a guide reports illness, a spare is reassigned immediately.

Guides receive their delegation's flight details, dietary notes, and any language context before arrivals day. On arrivals day, teams collect their goodie bags at check-in.

Critical notes:

- Because contestants are usually minors (as defined by the host country's law), all guides working with them must hold appropriate background checks as required by the host country's national law. Background checks must adhere to the law of the host country, not that of the delegation country, which would be impossible for a host to arrange.
- All members of the guide corps handling communications with foreign participants must be fluent in English.

Key milestones:

- 3 months out: Recruit guide corps; assign Superguides.
- 6 weeks out: Briefing session 1: IOI overview, delegation model, cultural sensitivity, contestant welfare.
- 2 weeks out: Briefing session 2: delegation assignments, contest rules, communication protocols.

- Day –2: Full dry-run of the arrivals procedure.
- Day –1: Guides equipped with badges, shirts, and any other branded kit they need for the event.
- Day 1 morning: Goodie bags ready for team collection at check-in.
- Day 1: Team guides present at check-in to welcome their delegations. A guide presence at the airport is a nice-to-have but is not required; hosts that cannot resource it can rely on the Airport Coordinator’s volunteers at the airport and have the team guide meet the delegation at check-in instead.

6.2 Medical Coordinator

Reports to: HOC Chair

Headcount: 1 Coordinator + contracted medical staff (at minimum: 1–2 nurses on-site; 1 doctor on rapid-response standby)

Active from: 2 months before IOI

One or two nurses on-site throughout the programme, with a doctor reachable within minutes, is the baseline; on competition days the doctor must be in the same building as the contest hall. The doctor does not need to sit at the venue: if the venue is adjacent to a hospital or clinic, a direct arrangement with that facility is often more effective, and considerably cheaper, than keeping a doctor on-site all week. Medical staff also cover all excursions, and medical rooms are set up at both the contestant accommodation and the main IOI venue(s). Women’s sanitary products must be freely available at all venues, including inside the contest hall, and replenished daily.

The Medical Coordinator collects pre-declared medical conditions from registration, briefs the medical staff, and is the escalation point for any medical incident during the week.

In a medical emergency, guides and volunteers call medical staff directly; reaching medical staff must never depend on first routing a message through the operational management chain. The notification chain that keeps the management side informed runs in parallel, not as a prerequisite: guide → Superguide → Head of Guides → Medical Coordinator → HOC Chair.

Key milestones:

- 2 months out: Medical staff contracted; medical rooms identified.
- Day –1: Medical rooms fully set up and stocked at both the accommodation and the main venue(s).
- Day 1: Medical room open from first arrivals.
- Days 3 and 5: Doctor in contest building throughout competition.
- All excursion days: Medical coverage confirmed.

6.3 Safeguarding

Contestants are minors and the IOI brings participants together across a wide range of cultural and legal contexts, so safeguarding sits alongside medical support as a core Participant Care responsibility. Two IOI-wide instruments frame it: the IOI Code of Conduct (IOI, 2026a), which applies to everyone involved with the IOI, and the IOI Allies Programme (IOI, 2026b), which provides trained community members as a confidential, on-site first point of contact independent of the host. Hosts should support both: brief all staff and volunteers on the Code, make it available to every delegation on arrival, and ensure that every participant knows the Allies Programme exists and how to reach it.

Beyond that, the host is responsible for drawing up and enforcing its own safeguarding rules appropriate to the local legal and cultural context. This paper does not prescribe those rules; what matters is that the host thinks them through explicitly before the event, writes them down, and briefs everyone in contact with participants on them.

7. Arrivals, Departures, and Airport

This domain manages all participant movement: inbound on Day 1, outbound on Day 8, and the bus logistics that connect both to the venue. Four coordinator roles cover the domain. The **Arrivals Coordinator** and the **Departure Coordinator** are peers reporting directly to the HOC Chair; in practice these two are usually filled by the same person, since both revolve around airport logistics and bus scheduling, albeit on different days. Two further roles sit under the Arrivals Coordinator: the **Check-in Venue Coordinator** runs the on-site reception point where arriving delegations receive their badges and goodie bags, and the **Airport Coordinator** owns the arrival points themselves.

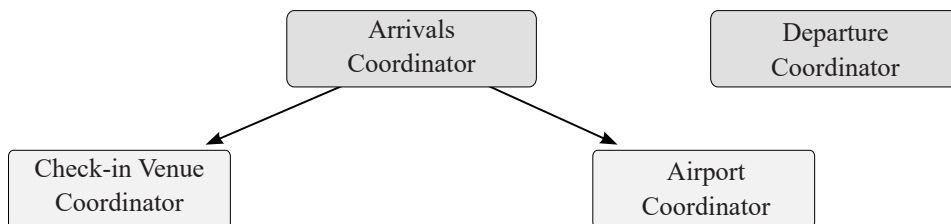


Figure 2: Roles within Arrivals, Departures, and Airport. The Arrivals Coordinator and Departure Coordinator are usually the same person.

7.1 Arrivals Coordinator

Reports to: HOC Chair

Headcount: 2 Coordinators + 1 volunteer assistant

Active from: 3 months before IOI

Arrivals day is defined by uncertainty: flights arrive late, connections are missed, delegations appear at the wrong terminal, and a handful of teams will reroute entirely in the 24 hours before landing. The Arrivals Coordinator's job is to absorb that uncertainty so that it does not propagate into the rest of the event. The day itself is a roughly 30-hour non-stop operation from the first early-morning flight until the last bus reaches the hotel, and two coordinators share the shift with clear handover points and overlapping availability during peak hours.

The Arrivals Coordinator builds the **master arrival schedule** from data in the IOI Registration System, which contains contact information and travel details for every delegation. The schedule lists each team's flight number, estimated time of arrival, number of passengers, and any special requirements.

A **clear communication channel** must be established so that team leaders can report last-minute changes (delayed flights, missed connections, alternative routing) directly to the Arrivals Coordinator. This may be a dedicated phone number, e-mail address, Discord or Matrix channel, or a combination. The channel details are communicated to all leaders at least two weeks before the event.

On arrivals day the Coordinator tracks live flight data alongside incoming messages from leaders and maintains a running status list.

Critical notes:

- Arrivals can start well before 06:00 depending on international flight schedules. The Arrivals Coordinator must be available from the first expected arrival until the last bus on Day 1 and must not carry other duties on arrivals day.
- **Early arrivals.** Some delegations will arrive the day before Day 1, or occasionally even earlier, on cheaper flights or awkward connections. The host is not expected to run a pickup service outside arrivals day for them: the rule is that a delegation arriving before arrivals day is responsible for making its own way to a designated pickup point in time for a Day 1 pickup slot. Publish the pickup points, the Day 1 pickup window, and any guidance for delegations that may land earlier, well in advance.
- Be welcoming to all delegations, but the host determines the pickup and drop-off points. There is no obligation to accommodate individual requests for alternative locations or special transport arrangements. A clear, fair, and uniform policy avoids setting precedents that create escalating demands from other delegations.

Key milestones:

- 3 months out: Pickup points decided and published; coordinated with the Airport Coordinator's initial contact with airport authorities.
- 6 weeks out: Communication channel set up.
- Day -14: Communication channel details sent to all leaders.
- Day -2: Master arrival schedule finalised; distributed to Airport Coordinator, Transport, and Head of Guides.
- Day -1: Monitor for last-minute flight changes.
- Day 1: Live monitoring from first arrival until last bus.

7.2 Check-in Venue Coordinator

Reports to: Arrivals Coordinator

Headcount: 1 Coordinator + 10–15 volunteers on packing days + 6–10 volunteers on check-in day

Active from: 3 months before IOI (procurement); Day -2 (packing)

The Check-in Venue Coordinator runs the on-site check-in point at the venue where arriving delegations are received. The scope is intentionally narrow: the end-to-end check-in process (airport pickup, bus transport, handover, badge and bag collection, delegation arrival at the hotel) is owned by the Arrivals Coordinator; this role owns the physical venue at which delegations receive their badge and goodie bag, and the packing operation that feeds it. From the organisers' perspective, the primary deliverable is the **badge**: it allows staff to see at a glance who has access to which areas. From the participants' perspective, the main deliverable is the **goodie bag**. Every participant receives one on arrival, with the badge attached to the outside. Standard contents: lanyard, T-shirt in the correct size, IOI programme booklet, notebook, pen, national souvenirs, and any sponsor items.

The arrangement above is not the only valid approach. Another workable option is to separate the badge and goodie-bag hand-outs: the badge is issued from the check-in desk, while the goodie bag (labelled with the T-shirt size it contains) is collected from a neighbouring counter, which avoids having to match badges to specific bags during packing. Whichever approach is chosen should be settled well before packing starts, since it changes what gets packed and how bags are labelled.

T-shirt sizing is the primary complexity of this role. Sizes are collected through the registration system. The Coordinator exports a size breakdown per participant, orders accordingly, and keeps surplus stock of mid-range sizes (M and L) to absorb late registrations and sizing errors.

Packing begins two full days before arrivals day with a volunteer team of 10–15 people, working from the final registration export. Bags are sorted by delegation and clearly labelled. On arrivals day, each team collects their bags at check-in.

Critical note:

Badges must be printed and physically attached to the correct bag before Day 1. Distributing badges from a separate general desk creates delays and confusion when nearly all delegations arrive within a few hours of each other.

Key milestones:

- 3 months out: Finalise design of all swag items; place orders (allow at least 8 weeks for custom items).
- 6 weeks out: Confirm T-shirt sizes from registration; adjust orders.
- 6 weeks out: Confirm delivery schedule for sponsor items to the packing location; sponsor items often arrive late and require active chasing.
- Day -3: All items received; packing station set up.
- Day -2: Full-day packing session with volunteer team.
- Day -1: Packing complete; bags sorted by delegation; quality check done.
- Day 1 morning: Bags ready for team collection at check-in.

7.3 Airport Coordinator

Reports to: Arrivals Coordinator
Headcount: 2 Coordinators + 4–6 airport volunteers
Active from: 3 months before IOI

The Airport Coordinator is physically present in the airport arrivals hall for the full duration of arrivals day. Working from the master schedule provided by the Arrivals Coordinator, their function is handover: each arriving delegation is met, checked off the list, and directed to the waiting bus. Like the Arrivals Coordinator, the Airport Coordinator role cannot be covered by a single person: arrivals day is a roughly 30-hour non-stop operation, and two coordinators must share the shift with overlapping coverage during peak hours.

Some hosts will need to cover more than one arrival point. In Europe, for instance, many delegations arrive by train rather than by air, and in countries with multiple international airports a significant share of arrivals may route through a secondary airport. In these cases the Airport Coordinator role is effectively duplicated: one coordinator (plus volunteers) at each arrival point, all reporting to the Arrivals Coordinator, who holds the master schedule across all points.

The list of pickup points should be published well in advance, with clear guidance for any delegation arriving outside of them; in a large country, delegations can reasonably be asked to make their own way to a designated pickup point rather than expecting the host to meet them anywhere they happen to land.

Buses are *not* pre-grouped into fixed time windows. Instead, each bus departs once it is

close to full or when a natural gap in arrivals occurs, so that delegations are not kept waiting at the airport unnecessarily.

A tablet or printed sheet records the delegation name, expected versus actual number of arrivals, and time of handover. Any discrepancy (fewer people than expected, an unaccompanied minor, an unexpected additional person) is reported to the Arrivals Coordinator immediately.

Critical notes:

- IOI-branded signage, clearly visible from the international arrivals exit and displaying a contact phone number, must be agreed with the airport authority in advance and installed before the first expected arrival. This requires a formal site visit, not an assumption.
- If the host country offers visa on arrival, try to arrange for at least two volunteers to be present behind security, near the visa-on-arrival counter. These volunteers should know the visa rules, speak the local language, and be easily recognisable (IOI-branded clothing or vests). Arriving delegations unfamiliar with the local process benefit enormously from a friendly face who can help navigate the paperwork. Always send volunteers in pairs, never alone.

Key milestones:

- 3+ months out: Initial contact with airport authorities at every arrival point, including the visa-on-arrival counter where applicable. Access, signage, and volunteer placement behind security cannot be arranged at short notice.
- 3 weeks out: Site visit to each airport; agree signage location and access with the airport authority.
- Day -1: Briefing of all airport volunteers.
- Day 1: On-site from first expected arrival at every arrival point.

7.4 *Departure Coordinator*

Reports to: HOC Chair
Headcount: 2 persons
Active from: 1 month before IOI

Departures day mirrors arrivals in length (another 30-hour non-stop span, shared between two coordinators), but not in character. Where arrivals are defined by uncertainty, departures are defined by scheduling discipline: the schedule must be published, locked, and communicated in a strict sequence, and a late or loose schedule cascades into missed flights.

The Departure Coordinator extracts outbound flight information from the registration system and constructs the departure bus schedule. Buses are grouped into departure windows

that account for standard international check-in times (typically three hours before departure).

Schedule timeline. Although all leaders enter their departure details in the registration system well before the event, flight information frequently changes at the last minute; leaders often only discover rebookings or schedule changes on their way to the IOI. The Departure Coordinator works to the following timeline:

1. A **preliminary draft** of the departure schedule is ready before the IOI starts, based on the registration data available at that point.
2. The **final draft** is posted to all leaders on the evening of Day 5, immediately after Competition Day 2 concludes.
3. Leaders have until **the evening of Day 6** to raise corrections or request changes. After that the schedule is locked.

Because flight details change so often, expect a significant number of corrections between the preliminary draft and the final lock.

Critical notes:

- A late schedule cascades into underprepared buses, stressed leaders, and missed flights. The Day 5-evening posting is the minimum required to give the transport team sufficient time to reorganise.
- Drop-off points follow the host-defined policy established for arrivals; individual requests for alternative locations need not be accommodated.

Key milestones:

- Day 5 evening: Draft schedule posted to all leaders.
- Day 6 evening: Change window closes; schedule locked and distributed.
- Day 8: On-site coordination of departures.

8. Venues and Facilities

This domain covers the physical preparation and in-event operation of all spaces used during the IOI week: the contest hall, the GA meeting room, the Fun Zone, and all signage and branding across every venue.

Role: Venues & Facilities Coordinator
Reports to: HOC Chair
Headcount: 1 Coordinator
Active from: 10 months before IOI

The Venues & Facilities Coordinator is the single point of responsibility for every physical space used during the IOI, from the contest hall through to the ceremony venues and the hotel lobbies where branding is installed. The role does not run any of the venues

directly; each venue has its own coordinator who owns the day-to-day details. What this coordinator does is hold the full picture across venues, resolve conflicts between them (for example between Fun Zone sound and contest hall quarantine, or between catering service points and branding placement), and act as the escalation point when a venue-level issue cannot be resolved within one sub-team. The Catering Coordinator sits inside this domain as well, because meals are organised around venue layouts, service capacity, and the daily programme. The five sub-leads (Contest Hall, GA Room, Fun Zone, Branding, Catering) all report to this coordinator.

8.1 Contest Hall Coordinator

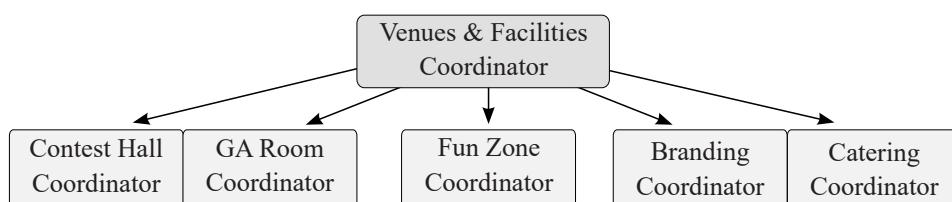


Figure 3: Roles within Venues and Facilities.

Reports to: Venues & Facilities Coordinator

Headcount: 1 Coordinator + sub-teams for networking, power, and physical setup + a pool of setup and contest-day volunteers

Active from: 10 months before IOI

The Contest Hall Coordinator is the single operational point of contact with the ITC throughout the setup period and during the event. They are usually a member of the HTC, or at least in very close contact with the HTC. It is quite normal for different people or contractors to be responsible for layout and physical setup, networking, and power; they should all report to the Contest Hall Coordinator.

Ideally, the contest hall should be ready four to five days before arrivals day, leaving enough time to source replacement equipment if anything is broken or missing. In practice, venues are rented by the day, so setup usually starts two to three days before arrivals day and all testing is compressed into the day before arrivals. HTC and ITC members should be involved on-site from the start of setup; ensure that catering is arranged for them, as they will be working long days. Schedule more volunteers than you think you need during setup: it is better to have a volunteer sit and read a book for a while than to be one person short when something needs carrying, cabling, or fixing.

Critical notes:

- The ITC inspects the technical setup on-site before the practice session. Budget time in the schedule to act on any findings, since passing the inspection is a precondition for the practice session.

- One or two ITC representatives should be on-site one to two days before the practice session. Final issues are almost always found in the last 24–48 hours before Day 2, and having the ITC physically present during that window drastically shortens the feedback loop.
- Every critical system function must be covered by at least two qualified individuals. Single points of failure in technical administration are unacceptable at the scale of the IOI.
- **Toilet plan.** A toilet plan must be prepared before the contest. Bathrooms must be close enough that contestants do not lose significant contest time walking to and from them. Toilets for every participating gender must be available at roughly equal distance from the contest hall, so that no group of contestants loses more contest time than another. Restroom visits must also be supervised or structured so that contestants cannot communicate with each other or access unauthorised materials. The plan should specify locations, supervision arrangements, and the procedure contestants follow to leave and re-enter the contest hall.
- **Paper and pencils.** An ample supply of blank paper and pencils must be available at every workstation. Many contestants work through problems on paper before coding.
- **Request signs.** Each contestant receives a set of printed signs they can hold up to request assistance: one each for snacks, bathroom, printer, and paper. Signs must be printed doublesided so that supervising volunteers can read them regardless of which way the contestant holds them up. Use a distinct colour for each request type so that volunteers can identify the request from across the hall without needing to read the signs. Prepare one full set per contestant.
- **Snack logistics.** Water and light snacks are provided to contestants during each five-hour contest (see Section 8.5). The logistics of distributing these snacks must be thought through carefully: how and when are snacks placed at workstations, how are they replenished, and how is this done without disturbing contestants? A plan for snack distribution should be part of the contest hall operational plan.
- **Security.** The contest hall is filled with expensive equipment: hundreds of workstations, networking gear, servers, and monitors. This equipment must be secured at all times, both during setup days and overnight between contest days. Access to the contest hall outside of official hours must be restricted and monitored. Ensure that the venue provides adequate security, and that the hall can be locked when not in use.

Key milestones:

- 10+ months out: Network and power specifications submitted to ITC for review. (The contest hall itself is reserved earlier, in the 24–12 months window, together with the hotels.)
- 5 months out: Workstation procurement finalised.
- 6 weeks out: Full network load-tested.
- Day -3: Full physical setup complete.

- Day -1: Dress rehearsal: full system test.
- Day 2 (Practice Session): Live trial; all issues resolved before Competition Day 1.

8.2 GA Room Coordinator

Reports to: Venues & Facilities Coordinator

Headcount: 1 Lead + 1 A/V volunteer + 3–4 printing volunteers during translation nights

Active from: 2 months before IOI

Seven General Assembly meetings take place during the IOI week. These meetings are chaired by the **Chair of GA**, an independent individual selected by the host country in cooperation with the IC (Regulation S3.3 (IOI, 2021)). They must have good English communication skills and extensive experience in chairing meetings. The host typically introduces the suggested GA Chair to the IC during the winter meeting preceding the IOI. The GA Chair is not part of the HOC; the GA Room Coordinator and the GA Chair should coordinate closely, since one owns the room logistics and the other presides over the meetings.

In principle, the GA room also serves as the venue for the two translation nights, and this paper therefore refers to the single GA room throughout. In practice, however, it often happens that the first GA meeting and the final GA meeting are held in a different room, for example a hall adjacent to the ceremony venue. Hosts should plan for this possibility while keeping the main GA sessions and both translation nights in one and the same room.

The GA Room Coordinator ensures the room is fully prepared at least 30 minutes before each GA session. The requirements group into five areas.

A/V.

- **High-quality audio.** The room must have good acoustics and a professional sound system so that leaders can hear each other clearly during debates and votes. Several floor microphones (not just one) must be available, so that multiple leaders can queue to speak without bottlenecking the debate. Poor audio leads to repeated questions and misunderstandings.
- **Projection.** A projector or large display (or a small number of them distributed around the room) visible to the entire audience, including from the back and when people are standing or walking around.

Network and power.

- **Reliable Wi-Fi.** Do not assume the venue's existing Wi-Fi is adequate. There will be 200+ people working for many hours, each bringing several connected devices, requiring the network to handle 400+ devices simultaneously. The Wi-Fi must be explicitly tested under realistic load conditions well before the event, and the connection must be guaranteed by the venue or a dedicated network provider. Translation

nights are particularly demanding: leaders use bandwidth-heavy web applications for hours on end, and a network failure during translation can delay the entire contest schedule.

- At least two power outlets per delegation so leaders can plug in their laptops.

Delegation identity.

- Country name signs and voting sticks at the correct places, verified against a printed delegation list before each session.
- Pigeonholes, one per delegation and one per IOI committee member. The IC, ISC, and ITC distribute materials through them, and leaders use them in both directions: to collect schedules, certificates, and announcements, and to drop off small gifts, notes, or items exchanged between delegations. Pigeonholes must be accessible around the clock for leaders and committee members.

Printing and storage.

- **Multiple high-load printers** accessible in or near the GA room. During translation nights every task is printed multiple times for each delegation, so a single consumer-grade printer, or even a few of them, will not cope.
- Dedicated storage space near the GA room for printed tasks and other materials.

Comfort.

- Coffee, tea, water, and light snacks at every session.
- **Climate control** must work reliably, including deep into the night during translation sessions. Leaders working through the night must not be left in a room that is either too hot or too cold because the building's HVAC system switches off after business hours.

An A/V volunteer operates under the GA Room Coordinator's direction and owns the projection and audio equipment specifically. The GA Room Coordinator is generally not a member of the HTC, but must be in close contact with them regarding Wi-Fi infrastructure and the translation systems used during translation nights.

As with the contest hall, the equipment and technical setup of the GA room are subject to ITC approval. The GA Room Coordinator should involve the ITC early in the planning process to ensure the room meets the required specifications.

The GA Room Coordinator is also responsible for transferring the printed tasks from the GA room to the contest hall so that everything can be placed on contestants' desks before each contest starts.

Key milestones:

- 2 months out: GA room confirmed; Wi-Fi, power, and A/V specifications agreed with venue and reviewed by ITC.
- 6 weeks out: Full Wi-Fi load test under realistic conditions; printer and paper supply confirmed.

- Day -2: Room fully set up: country signs, voting sticks, pigeonholes, power, printers, A/V.
- Day -1: Dry-run of projection, audio, and printing workflow.
- Day 2: Room ready 30 minutes before GA 1.
- Translation nights (Days 2 and 4): Overnight readiness of printers, paper, Wi-Fi, climate control.

8.3 Fun Zone Coordinator

Reports to: Venues & Facilities Coordinator

Headcount: 1 Coordinator + 3–4 facilitators per open session

Active from: 5 months before IOI (layout); 6 weeks before IOI (content)

The Fun Zone is a recreational area for contestants. To keep contestants close to the rest of the programme and to minimise additional transport, the Fun Zone should be located as close to the competition venue as possible, ideally within walking distance or in the same building complex.

The Fun Zone is typically open at three moments during the week: after Competition Day 1 (Day 3 afternoon), after Competition Day 2 (Day 5 afternoon), and potentially on the morning with the tech talks (Day 7). The Day 7 session is the hardest to realise: by Day 7, deconstruction of the competition infrastructure is usually already underway, and the Fun Zone venue (if co-located with the contest hall) may not be available. Hosts should plan for this and either decouple the Fun Zone venue from the contest hall for Day 7, or accept that Day 7 may not feature a Fun Zone session.

The Fun Zone Coordinator manages all logistics and coordinates with the Branding Coordinator for signage and sponsor placement within the space. Sponsors will place booths in the Fun Zone, so the Fun Zone Coordinator must confirm the **layout at least five months in advance** to give sponsors time to design their booths. The exact content of games and recreational activities should be finalised six weeks beforehand.

Facilitators are usually teachers. Their role is to help contestants interact with each other and provide a safe, fun, and relaxing environment. Contestants should never be forced to participate in activities. A dedicated **quiet zone facilitator** should also be available for contestants who prefer a calmer space.

Booth space and visibility for long-term sponsors are agreed with the IOI Treasurer (see Section 3.4).

Critical notes:

- If there is no sound barrier between the Fun Zone and the contest hall, no construction or activity can take place in the Fun Zone during contests. All setup and construction must therefore be completed before Competition Day 1.

- Sponsors sometimes escalate their requests as the event approaches, asking for more space, more prominent placement, or activities that interfere with the programme. It is acceptable, and sometimes necessary, to say no. The wellbeing and enjoyment of the contestants come first. The Fun Zone exists for the contestants, not as a trade fair.

Key milestones:

- 5 months out: Fun Zone layout confirmed; communicated to sponsors.
- 6 weeks out: Games and recreational content finalised.
- 3 weeks out: Facilitators recruited and briefed.
- Day -1: Fun Zone fully set up and tested.
- Days 3, 5, and 7: Facilitators present during open sessions.

8.4 Branding Coordinator

Reports to: Venues & Facilities Coordinator

Headcount: 1 person

Active from: 3 months before IOI

The Branding Coordinator is responsible for all printed signage across every IOI venue: contest hall, GA room, Fun Zone, check-in area, hotel lobbies, airport meeting point, and ceremony venues.

Sponsor visibility is only one part of this role. Just as important is that participants feel welcomed and part of the IOI community from the moment they arrive. That means the current year's IOI logo and visual identity should be clearly and consistently visible throughout every venue, so that contestants, leaders, and guides all know they are at the IOI rather than at a generic conference. Branded IOI backdrops and banners should also be placed in photogenic spots at the main venue(s), hotels, and ceremony areas, so that participants can naturally take selfies and group photos against them. These photos travel back to home countries and are a large part of how the event is remembered.

The sponsor hierarchy (diamond, gold, silver, bronze) determines the relative sizes of sponsor logos on banners and other printed material, not the sizes of the banners themselves. This hierarchy must be confirmed in writing before any designs are finalised, and all logo files must be obtained in vector format. The Branding Coordinator maintains a master inventory recording what banner goes where, at what dimensions, and who installs and removes it.

The Branding Coordinator also coordinates with external vendors that deliver booths for the IOI Fun Zone and other sponsored areas, ensuring that booth designs and dimensions align with the approved layout.

Although the Branding Coordinator reports within Venues & Facilities, the role sits at the intersection of venues, sponsorship, and external communication, and draws on three

working relationships outside the venue domain: the **IOI Treasurer** on long-term IOI-wide sponsors, the host's **sponsorship team** on host-specific sponsors for this edition, and the **Press & Media Lead** on sponsor tagging for social media and other external channels (see Section 3.4). The Branding Coordinator does not own sponsor relationships, but functions as the single source of truth for how every sponsor is visible at the event.

Critical notes:

- Sponsor visibility is a contractual obligation. A missing or incorrectly sized sponsor banner is a serious issue. All designs must be approved before going to print.
- Remember that the host is in charge. The IOI is an event for the contestants, not for the sponsors. The children and the competition come first. Sponsor requests that conflict with the programme, the welfare of participants, or the integrity of the competition must be declined.

Key milestones:

- 3 months out: Confirm sponsor list and sizes; receive all logo files; confirm placement of IOIbranded backdrops and selfie banners at main venue(s).
- 2 months out: All designs approved; go to print.
- Day -2: All IOI week banners installed.

8.5 Catering Coordinator

Reports to: Venues & Facilities Coordinator
Headcount: 1 Coordinator + contracted catering vendor
Active from: 3 months before IOI

The Catering Coordinator provides meals for approximately 700 accredited participants (contestants, leaders, deputies, guides, committee members, and host organisers), plus the additional volunteer pool, across the full IOI week. Dietary requirements exported from the registration system are provided to the catering vendor at least six weeks before the event.

Although catering sits within Venues and Facilities because meals must be organised around venue layouts, service capacity, and the daily programme, it is in part a participant-care function: individual dietary needs, allergies, and declared medical conditions all flow in from the registration data and ultimately concern the wellbeing of each participant. The Catering Coordinator must therefore work closely with the Registration Lead (for dietary counts and allergy declarations) and with the Medical Coordinator (for participants with medical conditions that affect what they can eat), and should think of the role as serving both the venue schedule and the people inside it.

Labeling and separation. All food must be clearly labelled in English, with icons indicating: Halal, Vegetarian, Vegan, Chicken, Fish, Pork, Beef, Nuts, Gluten, and Lactose. Food

categories should be physically separated to minimise the risk of cross-contamination. In practice, most meals will see a large share of participants arrive within a relatively narrow window, so multiple food stations must be set up to avoid queues. Clean, fresh drinking water must be available in unlimited supply for all participants at all times and at every venue. Other beverages (juices, soft drinks) are a welcome addition.

The Catering Coordinator must coordinate with the Competition Operations team so that lunch timing can be adjusted if a contest starts or finishes late.

Several event-specific requirements must be explicitly planned for:

- **Competition day breakfasts** must open early enough for all contestants to eat and still be seated in the contest hall at the contest start time. Allow a minimum of 45–60 minutes between breakfast opening and the first contestant seated. Even large hotels tend to struggle with the IOI because contestants need to eat within a narrow window before the contest starts. After a contest, large numbers of participants arrive together and want to eat at roughly the same time. This must be discussed explicitly with the catering vendor and hotel management well in advance.
- **Translation nights:** Buffet food must be available throughout the night in the GA room. Do not assume all leaders finish by midnight; some will still be working at breakfast time. It is important that leaders can eat quickly before translations begin so the session does not start late. The exact menu is for the host to decide; as a suggestion, coffee, tea, water, soft drinks, and savoury and sweet snacks work well, and hot items such as soup or light hot meals deep into the night are appreciated when the weather or the schedule call for them.
- **Contest hall:** Water and light snacks must be available for contestants throughout each five-hour contest. Snacks should be multiple small items that do not create litter, and clearly labelled as sweet or savoury, gluten-free, and nut-free. It is recommended that all contest snacks are halal, vegetarian, vegan, and kosher, so that the only remaining concerns are gluten and nut allergies. Thought should also be given to the noise factor: the contest hall is a quiet zone, so avoid snacks that are loud to eat or use wrapping that is noisy to open.
- **Fun Zone:** More drink options (beyond water) should be provided during Fun Zone open sessions.
- **Arrivals day (Day 1):** Food must be available until very late at night, well beyond normal dinner hours. Many delegations arrive on late flights and reach the hotel or venue after the regular dinner service has ended. If the catering vendor cannot provide a full hot meal that late, boxed dinners are an acceptable fallback, but food *must* be available.
- **Departures day (Day 8):** The mirror image of arrivals day. Delegations depart across a 30-hour window, with the earliest buses leaving in the middle of the night. Breakfast and snack catering must be continuously available from before the first bus until after the last departure; boxed breakfasts for participants whose bus leaves before the buffet opens are a reasonable solution. This is commonly overlooked and leaves tired travellers with nothing to eat on the way home.

- **Awards banquet (Day 7):** The awards banquet follows the Closing Ceremony and is the last shared meal of the IOI. It does not have to be a seated dinner: a standing reception, buffet, or food-station format works equally well. Whatever format is chosen, it must feel special and celebratory. Waiting times should be kept short. 700 participants arriving at the same time means the food service needs to be fast and well-organised, and self-service stations are a common and effective way to achieve that. A seated format with individually plated courses is very difficult to time well for 700 people simultaneously; long waits between courses, or before the first course, leave a negative final impression of the entire event.
- **Alcohol:** Many participants are underage or come from cultures where alcohol is not permitted. Avoid alcoholic beverages as much as possible, and ensure that glasses typically associated with alcohol (wine glasses, champagne flutes) are not used to serve non-alcoholic drinks. Photos do not show the contents of a glass, and some participants may face serious consequences at home.
- **Hotel minibars** in *every* participant room, including leader rooms, should be locked or emptied before check-in. The host does not want to be invoicing individual delegations for minibar consumption at check-out. Coordinate with each hotel well in advance.

Key milestones:

- 3+ months out: Catering vendor contracted.
- 6 weeks out: Final dietary counts submitted to vendor.
- Day 1: Arrivals day catering for all participants, continuously available until well after the last delegation has reached the hotel.
- Days 3 and 5: Early breakfast service; snacks in contest hall.
- Days 2 and 4 (evenings): Overnight buffet in GA room.
- Day 7: Awards banquet.
- Day 8: Departures day catering, from before the first bus until after the last departure (boxed breakfasts for early buses).

9. Competition Operations

Competition Operations covers the operational interfaces between the HOC and the scientific/technical committees. The HTC is responsible for the entire technical part: network, power, infrastructure, printers, machines, the contest management system, the translation system, and related systems. The HSC is responsible for the tasks, test data, and graders. The HTC and HSC are also very much involved during the translation nights: they run and maintain the translation system infrastructure and generally run those two evenings. The HOC's role is to ensure that the physical and logistical infrastructure needed by those committees is in place at the right time. Three coordinator roles sit at this interface.

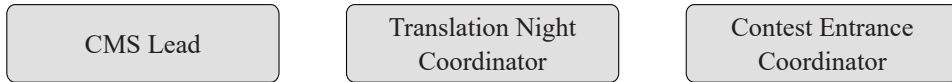


Figure 4: Roles within Competition Operations.

9.1 CMS Lead

Reports to: HOC Chair (coordinates with ITC)

Headcount: 1 Lead + 2–3 technical staff on duty during all contest events

Active from: 6 months before IOI

The CMS Lead is a member of the HTC and is responsible for installation, configuration, network integration, and testing of the Contest Management System. The ITC must approve the configuration. A testing installation of CMS must be operational six months before the IOI so that the ISC can verify that the host is able to install and run the system correctly. Two staff members must be on duty in the server room at all times during the practice session, both competition days, and both translation nights. The Lead manages the public scoreboard and confirms with the ITC whether live scoring will be publicly accessible.

The practice session on Day 2 is the final opportunity to identify and resolve system issues before Competition Day 1. Any problem found during the practice session must be fixed before the following morning.

Key milestones:

- 6 months out: Testing installation of CMS operational; reviewed by ISC and ITC to confirm the host can run the system.
- Day –2: Full system test with load simulation.
- Day 2 (Practice Session): Live trial; fix all issues before Competition Day 1.
- Days 3 and 5: Two staff in server room at all times.

9.2 Translation Night Coordinator

Reports to: HOC Chair (coordinates with ISC)

Headcount: 1 Coordinator + 3–4 volunteers for print and distribution

Active from: 1 month before IOI

Two translation nights take place during the IOI week: on the evenings before Competition Days 1 and 2 (Days 2 and 4 respectively).

Translation nights take place in the GA room (see Section 8.2). Three roles converge on this activity and it is worth being explicit about the boundaries: the GA Room Coordinator

is responsible for maintaining a working room (audio, seating, power, climate control); the Catering Coordinator owns all food and drink for the translation nights as part of the overall IOI catering plan (see Section 8.5); and the Translation Night Coordinator owns the translation process itself, including printing and paper supplies, task distribution, and coordination with the ISC. Overnight catering is therefore *not* owned by the Translation Night Coordinator: they brief the Catering Coordinator on the expected duration and intensity of each translation night, but the food, drink, and service remain the Catering Coordinator's responsibility.

Sponsor branding is welcome in the GA room, but sponsors themselves cannot be present during translation nights. Translation nights are part of the quarantine period: only registered team leaders and authorised IOI committee members may be in the room while tasks are being translated.

All requirements for the GA room (Wi-Fi capacity, display, power, climate control) apply equally during translation nights. The Coordinator ensures the room is additionally equipped with one work desk (two seats, two power outlets) per delegation, and multiple printers. Overnight food and drink service is arranged through the Catering Coordinator. Delegations must bring their own laptops.

They manage the distribution protocol: how draft problem statements are released, how final versions are distributed, and how sealed printed statements are transferred securely to the contest hall.

Leaders should be briefed on the protocol at GA 1. Transport must be arranged for leaders leaving the GA room at any hour of the night.

Critical note:

Translation nights routinely extend until early morning. Catering and transport must remain available until at least breakfast time. Planning that assumes all leaders finish by midnight will leave some without food or transport.

Key milestones:

- 1 month out: GA room confirmed and fully equipped for translation nights.
- Day 2 evening: Translation Night 1 (before Competition Day 1).
- Day 4 evening: Translation Night 2 (before Competition Day 2).

9.3 Contest Entrance Coordinator

Reports to: HOC Chair (coordinates with ISC)
Headcount: 1 Coordinator + 2–4 volunteers at contest hall entrance
Active from: 2 weeks before IOI

The Contest Entrance Coordinator manages the flow of contestants into the contest hall before each contest and the practice session. Contestants must not bring bags, phones, smartwatches, or other prohibited items into the hall. The list of prohibited items is defined in the contest rules, which are approved by the ISC. On-site approval of actual items brought by contestants is performed by the ITC, who typically delegates this responsibility to the Contest Entrance Coordinator. The Coordinator translates these rules into a practical checkpoint process and briefs the guides stationed near the entrance to keep entry smooth and calm.

Even with clear communication beforehand, some contestants will arrive with prohibited items. To avoid unnecessary stress, the entry team must be prepared to collect and securely store these items and return them when the contest ends. A simple labelled-bag system (one per contestant) works well.

Key milestones:

- 2 weeks out: Prohibited items list confirmed with ISC; storage procedure designed.
- Day –1: Entry volunteers and guides briefed on procedure.
- Day 2 (Practice Session): Live trial of entry process.
- Days 3 and 5: Full entry checks before each contest.

10. Communications and Registration

This domain covers everything informational that flows in and out of the host: the data collected from every participating country, the day-to-day communication with leaders and delegations, and the external communication aimed at press, sponsors, and the wider public. Three roles sit alongside one another, each owning a distinct audience and channel. The Registration Lead owns the authoritative participant data and the visa support that flows from it. The Internal Communications Coordinator owns the channels aimed at the IOI community itself (leaders, contestants, committee members) and the participant-facing parts of the host website. The Press & Media Lead owns everything aimed at audiences outside the IOI community, including social media, press releases, and the ceremony livestream. All three report directly to the HOC Chair.

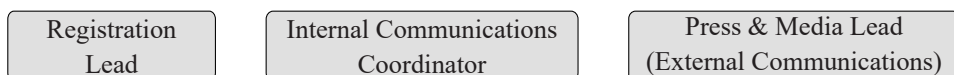


Figure 5: Roles within Communications and Registration.

10.1 Registration Lead

Reports to: HOC Chair

Headcount: 1 Lead + 1–2 visa support volunteers

Active from: 4 months before IOI

The IOI uses the IOI Registration System (<https://ioiregistration.org>), which is maintained internationally. The Registration Lead uses this system to manage all foreign participants, including IOI committee members. Invitation letters are generated from this system; countries that require a government-issued letter must be handled via the Ministry of Foreign Affairs.

Visa processing can take up to 30 days. For each country requiring a visa, the Lead identifies the relevant consulate, initiates contact, and monitors progress. The host should pursue visa-free arrangements for all IOI participants where possible.

The registration export is the authoritative data source for the entire operational team. It feeds:

- T-shirt sizes → Check-in Venue Coordinator;
- dietary requirements → Catering Coordinator;
- arrival and departure flight data → Arrivals Coordinator and Departure Coordinator;
- room-sharing preferences and partnered participants → Accommodation Coordinator;
- declared medical conditions → Medical Coordinator.

Critical notes:

- Registration must be actively chased. Many countries will not register without a reminder. A firm deadline must be set and enforced; late registrations create cascading problems for badges, bags, accommodation, and catering.
- All countries must register through the official IOI Registration System. No country may sign up or be added outside of this system. Any new participating countries or observer countries must first be approved through the IOI Secretary before they can be registered. The Registration Lead should direct all such requests to the IOI Secretary and not handle them independently.

Key milestones:

- 4 months out: Registration open; invitations sent to all countries.
- 3 months out: First registration chase.
- 2 months out: Visa support processed for all requesting countries.
- 6 weeks out: Export final dietary and T-shirt size data to relevant leads.
- Day –3: Final registration report; badges printed.

10.2 Internal Communications Coordinator

Reports to: HOC Chair
Headcount: 1 Coordinator
Active from: 2 months before IOI

The Internal Communications Coordinator is responsible for all communication aimed at the IOI community itself: team leaders, deputy leaders, contestants, and IOI committee members. This covers digital channels used in the run-up and during the event (Matrix rooms, Discord servers, email inboxes, and any other platforms the host decides to use), and the participant-facing parts of the host website. The role is not to answer every incoming question, but to make sure that every message is seen and routed to the right coordinator or lead without delay.

At the time of writing, Matrix is the IOI's official communication platform. The host's primary channel should preferably be a Matrix room on the IOI Matrix infrastructure, ensuring continuity with the platform that leaders and committee members already use. In practice, team leaders and deputies often also communicate on other channels: WhatsApp groups, Discord servers, email threads, or informal social media groups that form before and during the event. The host has two legitimate options here: either monitor (or delegate monitoring of) the most active unofficial channels, or make it explicit from the start that unofficial channels will not be listened to and that official questions must be raised through official channels. Either approach works, but the choice must be communicated clearly to all leaders well before the event.

During the IOI week, questions and requests arrive continuously on multiple channels. Without a dedicated person monitoring these channels, messages get lost, questions go unanswered, and participants lose trust in the organisation. The Internal Communications Coordinator triages incoming messages, delegates them to the appropriate operational lead, and follows up to ensure a response is sent.

An important consideration for all IOI communication: many participants and most organisers are not native English speakers. Important information should therefore always be communicated in written form. Relying solely on spoken announcements risks misunderstanding; written messages can be re-read, translated, and forwarded within delegations.

IOI host website. The Internal Communications Coordinator also owns the IOI host website. A first version showing the dates of the IOI and airport information should be online before IOI $n-1$ (the edition preceding the one being organised).

In the months leading up to the IOI, the audience of the website is explicitly the IOI community: participants, leaders, deputies, contestants, and the parents of contestants. They need practical information about travel, visas, schedule, competition rules, and the grading environment, and the website content should be written with that audience in mind.

Once the event has started, the website additionally serves the wider community (press, public, sponsors, alumni), and carries live information of public interest: scoreboard links, published tasks, and daily updates. The website should be updated at least daily during the IOI week. A link to the live scoreboard should appear immediately before each contest day, and the full text of all tasks and translations should be published immediately after each competition day ends. Publication of any contest-related material (scoreboard links, tasks, test data) must be approved by the ISC before it goes live. Items that primarily target the outside world (press announcements, social media posts, highlight coverage) are forwarded to the Press & Media Lead.

Key milestones:

- Before IOI $n-1$: First version of host website online with dates and airport information.
- 6 months out: Website updated with schedule, visa guidelines, competition rules, and grading environment.
- 2 months out: Communication platforms selected and configured (Matrix rooms on IOI infrastructure, dedicated email addresses).
- 6 weeks out: Channel details communicated to all team leaders and IOI committee members.
- Days 1–8: Continuous monitoring and triage of all channels; daily website updates including scoreboard links and task publications (with ISC approval).

10.3 Press & Media Lead (External Communications)

Reports to: HOC Chair

Headcount: 1 Lead + 2 writers/photographers + 1 livestream technician (optional)

Active from: 3 months before IOI

The Press & Media Lead owns all communication aimed at audiences outside the IOI community: general press, the wider public, sponsors, alumni, and anyone following the IOI from the sidelines. The main channels for which they are responsible are social media, press releases, local media coverage of ceremonies, and the public-facing parts of the host website (which are owned by the Internal Communications Coordinator, with content relevant to the outside world fed over by the Press & Media Lead).

Social media. Social media has become the primary external communication channel for recent IOIs. The Press & Media Lead maintains the event’s social media presence before, during, and after the IOI, with daily posts covering arrivals, ceremonies, contest days, excursions, and the closing events. Posts must be coordinated with the Branding Coordinator: sponsors have contractual visibility requirements that extend to social media, and all official posts must tag the relevant sponsors according to the agreed sponsor hierarchy. The Branding Coordinator maintains the list of sponsor handles and tagging requirements.

Daily newsletter (optional). Historically, hosts produced a printed daily newsletter distributed at breakfast. Recent IOIs have largely moved away from this: printing several hundred colour copies each morning is expensive and much of the audience now prefers digital updates. A host may still choose to produce a daily newsletter, printed or digital, and the option remains entirely reasonable, but it should be treated as one possible format among many rather than a required deliverable.

Livestream. Two distinct kinds of livestream should be considered separately.

The *contest livestream* covers the competition days (live scoreboard, cut-ins from the contest hall, commentary) and is what the IOI community has traditionally called “the livestream”. It is entirely optional, and most hosts should not attempt it: producing it well requires broadcastgrade capabilities on top of the contest data feed, and at recent IOIs it has only been realised when a dedicated community volunteer happened to be available to run it together with a local production company. In practice, hosts have found the effort-to-reward ratio unfavourable unless such a volunteer is already in place.

The *ceremony livestream* covers the Opening and Closing Ceremonies. Technically this is much closer to a standard event livestream, and can usually be delivered by the same A/V contractor who runs the ceremonies themselves. For most audiences outside the IOI community, such as contestants’ families, schools, and general press, the ceremony livestream is actually the more important of the two: it is the moment the event is most watchable from the outside. Hosts should therefore plan to livestream the Opening and Closing Ceremonies even if they decide not to attempt a contest livestream.

Press on-site. Local media attending the ceremonies are managed by this lead, with a designated accreditation process and a dedicated media zone at the ceremony venues.

Photo and video material. The host should produce high-quality photography and video of the IOI week, both as a contemporaneous record and for the Closing Ceremony highlight reel and photo compilation. After the event, selected photo and video material should be published through appropriate channels: social media, specialised photo and video platforms, and selected items on the host website. Publishing the entire archive on the host website is neither necessary nor always desirable; not all material will be of suitable technical quality or appropriate for unrestricted publication, and curation is part of the job.

The Closing Ceremony highlight reel and photo compilation require the photography and video team to actively collect material throughout the week with those deliverables in mind, not only for daily coverage and the archive.

Key milestones:

- 3 months out: Social media presence set up; livestream arrangement confirmed (or decision taken not to livestream).
- 6 weeks out: Editorial and social media plan approved; newsletter format decided (if produced).
- Days 1–8: Daily social media coverage; daily newsletter (if produced).

- Day 7 morning: Highlight movies and photo compilation ready for Closing Ceremony.
- Post-IOI: Curated selection of photo and video material published via appropriate channels.

11. Ceremonies

Role: Ceremonies Coordinator

Reports to: HOC Chair

Headcount: 1 Coordinator + A/V and production team (may be contracted) + a team of ushers and floor volunteers for both ceremonies

Active from: 6 months before IOI

The Opening Ceremony on Day 2 and the Closing Ceremony on Day 7 require a venue with a minimum of 1 000 seats. At IOI scale this figure is a floor, not a target: once guides, host volunteers, local dignitaries, sponsors, and press are added to the roughly 700 participants, 1 000 seats fills quickly. Both ceremonies are among the most visible moments of the IOI and reflect directly on the host country. Both ceremonies require a team of volunteers for ushering, seating control, and managing the flow of participants on and off the stage; this volunteer team should be planned and briefed for both ceremonies, not just the opening.

Opening Ceremony (Day 2) includes the naming of individual delegations while their photos are shown on screen, welcome speeches, music, dance or other entertainment, and the official start of the competition. It is customary for the IOI President to give a short opening speech; the Ceremonies Coordinator confirms their slot in the running order and briefs them on timing well before the day.

Delegations like to sit together. With an audience of this size it is worth assigning seats per team in advance: it makes filling the room fast and predictable, and ensures that no participant is pushed away from their delegation just because the seats next to them have already filled up.

The ceremony should be supported by clear on-screen background presentation (names and titles of speakers, segment titles, delegation names) and by professional event moderation. A printed programme is optional: it is useful for some audiences, but most copies end up discarded at the end of the ceremony, and a well-run show with good background presentation does the same job without the waste.

Interpretation. The working language of the IOI is English, but host officials, ministers, and other dignitaries may speak in the host country's language, and local press coverage of the ceremonies is usually conducted locally. For any non-English speeches in the running order, the Ceremonies Coordinator arranges live interpretation or pre-translated scripts with on-screen subtitles into English, and a short English-language summary is made available for local press who do not cover the IOI in English.

Closing Ceremony (Day 7) includes the medal presentation and the IOI Flag Handover to the following year's host country. The IOI President oversees the Flag Handover and typically addresses the closing audience as well. A highlight video reel of the IOI week should be screened at the start of the ceremony.

Of all the group sessions during the week, the Closing Ceremony is arguably the most important: it is what the contestants have worked towards, and it is the moment they will carry home with them. Keep the focus firmly on the contestants. Speeches, sponsor moments, and dignitary appearances should be tightly controlled so that nothing competes with the medal presentation and the recognition of the participants themselves.

Guides are an integral part of both the Opening and Closing Ceremonies. By the end of the week they are usually considered part of the delegation by the teams.

Critical notes:

- Speeches at both ceremonies should be short, and speakers should be explicitly asked to tailor their remarks towards the contestants. The audience is predominantly international secondary-school-aged contestants and their coaches. At the Opening Ceremony many will be jet-lagged, and at the Closing Ceremony they will be eager for medals, not speeches. Presenters must be briefed accordingly.
- At the IOI, 50% of contestants receive a medal. This means a large number of contestants will be called forward during the Closing Ceremony. Medal winners should be placed in assigned seats in a smart order so they can be called forward efficiently. The seating plan for medal winners is a key deliverable. To keep the ceremony moving, contestants are called up in groups of 8–12 per group for bronze and silver medals. Only the final 3–5 medal winners (typically the top gold medallists) are called up individually.
- **Medal presenters:** Government officials, sponsors, and other dignitaries will typically want to hand out some of the medals. A plan must be prepared well in advance specifying which important persons will present which medal groups. This requires coordination with the Host Steering Committee, the host sponsorship team (for host-specific sponsors) and the IOI Treasurer (for long-term IOI-wide sponsors), and the Branding Coordinator, and the plan must be confirmed and rehearsed before the ceremony.
- Medals must be ready before the Closing Ceremony. The exact cut-off is not known until after the results are finalised. Always produce significantly more medals than the minimum estimate, and ensure that ties at the top are covered.
- The Closing Ceremony also hosts the Distinguished Service Award (DSA) if one or more are given that year. There can be more than one DSA recipient. Confirm with the IC well in advance how many DSAs will be presented and plan the ceremony segment accordingly.
- **Stage visibility.** Contestants on stage must not be blocked from view by other contestants. Some contestants will try to bring large props on stage: stuffed animals, mascots, sombreros, or large flags. These items must be taken from the contestant

before they walk on stage and returned afterwards. Flags should be draped around contestants' shoulders rather than held up in front of others.

Key milestones:

- 6 months out: A/V production team contracted. (The ceremony venue itself is reserved earlier, in the 24–12 month window, together with the hotels.)
- 1 month out: Running orders drafted and approved.
- Day –1: A/V and volunteer walk-through of Opening Ceremony.
- Day 2: Opening Ceremony.
- Day 7 morning: Medal winner seating plan finalised.
- Day 7: Closing Ceremony and IOI Flag Handover.

12. Excursions

Role: Excursions Coordinator

Reports to: HOC Chair

Headcount: 1 Coordinator; guides serve as on-site escorts

Active from: 3 months before IOI

The Excursions Coordinator organises excursions, arranges tickets and entrance logistics, and coordinates closely with the Transportation Coordinator (for buses) and the Catering Coordinator (for meals). If an excursion is running late (for example due to traffic) the Catering Coordinator must be informed immediately so that meals can be delayed accordingly.

IOI committee members do not always participate in excursions and other programme activities. The Excursions Coordinator should confirm attendance with committee members in advance to ensure accurate headcounts for transport, catering, and entrance logistics.

Two excursions take place during the IOI week.

Day 4: Small excursion. Must return to accommodation before dinner. Leaders face a full night of translation immediately after; an excursion that runs into the evening will leave them exhausted before a difficult night. The contestants also still have an entire second competition day ahead of them and many will be focused on that: the Day 4 excursion should be light enough that nobody arrives at Competition Day 2 tired or injured, and it should not include activities that contestants may reasonably feel are too risky or too exhausting the day before their second contest.

Day 6: Big excursion. A full-day event; no translation constraint applies. Separate programmes for adults and contestants are possible.

A head count must be taken before every departure and after every return. Guides do not leave any excursion site without confirming the full complement of their delegation.

Medical safety net. Excursions take participants away from the main venue where the medical room is staffed. The Excursions Coordinator must plan the medical support structure for each excursion in advance: which medical staff or first-aid trained volunteers will accompany the group, what the nearest hospital is at each excursion location, and how an emergency evacuation would work. This planning must be completed and reviewed with the Medical Coordinator well before the excursion days.

Key milestones:

- 3+ months out: Excursion routes confirmed; tickets and logistics arranged; bus requirements communicated to Transportation Coordinator. Group-rate tickets and entrance arrangements at popular sites often need to be booked earlier than this.
- 1 month out: Guides briefed on excursion content.
- Day 4: Small excursion (must return before dinner).
- Day 6: Big excursion.

13. Transportation

Role: Transportation Coordinator

Reports to: HOC Chair

Headcount: 1 Coordinator

Active from: 2 months before IOI

The Transportation Coordinator ensures that participants are where they need to be, every day of the IOI week. In many hosts the same bus contractor (and, in small teams, the same person) handles arrivals, departures, and daily transport, and this is entirely workable for the run-up and post-event phases. During the IOI week itself, however, the daily transport load is heavy enough that it benefits from dedicated focus: a single person trying to cover arrivals, departures, and the daily programme simultaneously will struggle on peak days. In practice, many hosts split the role during the event week even if it is held by one person before and after, and that is the arrangement recommended here.

The cheapest, simplest, and most reliable transport is no transport at all. Where the accommodation, contest venue, GA room, and ceremony venue sit on or near the same site, daily bus operations disappear almost entirely, and excursion programmes can be chosen to minimise additional bussing as well. Hosts should consider this early, at the venue-selection stage: saving transportation load is usually far more valuable than any gain from spreading venues across a city.

Daily responsibilities include coordinating buses between hotels and the competition venue, the ceremony venue, excursion departure points, and any other event locations. The Transportation Coordinator works from a master schedule that reflects the IOI programme and adjusts in real time when events run late, excursions are delayed, or ad-hoc transport needs arise.

Close coordination with the Excursions Coordinator and the Catering Coordinator is essential: if transport is delayed, downstream schedules for meals and activities must be adjusted accordingly.

Loading buses for large groups takes a surprising amount of time, and needs to be accounted for in every schedule. Be transparent about the exact time people are expected to be ready and the exact time buses are scheduled to leave. By departing on time from the very beginning of the week, you train the teams to be punctual.

The temperature on the buses should be friendly to the passengers. This may involve discussing with the drivers to start the climate control before the participants head back to the bus.

Bus boarding discipline. Guides must be actively involved in getting people on and off buses. Without a clear boarding plan, 700 participants will self-organise chaotically, leading to delays and lost delegations. The Transportation Coordinator must decide on a boarding policy beforehand: either buses are assigned to specific delegations, only complete teams accompanied by their guide may board a bus, or any other structured mechanism that can be monitored. Whichever approach is chosen, it must be communicated to all Superguides and guides and enforced consistently from Day 1. A word of caution, however: over-engineering an inflexible boarding system can cause as much chaos as having no plan at all. If the system is too rigid (for example requiring every delegation to queue at a specific bus in a specific order), any small disruption (a late delegation, a changed bus) cascades into delays for everyone. The best boarding plans are simple, easy to communicate, and flexible enough to absorb the inevitable deviations of a live event.

Key milestones:

- 2 months out: Bus company contracted; route plans drafted.
- 1 month out: Daily transport schedule drafted based on IOI programme.
- Day -1: Final transport schedule distributed to all coordinators and guides.
- Days 1-8: Daily transport coordination.

14. Accommodation

Role: Accommodation Coordinator

Reports to: HOC Chair

Headcount: 1 Coordinator

Active from: 24 months before IOI (hotel inventory secured in the 24-12 month window, together with the venues)

The Accommodation Coordinator manages all hotel bookings and room assignments for approximately 700 participants. Room-sharing preferences and partnered participants are exported from the registration system and must be honoured where possible.

As discussed in Section 4, hotel inventory is one of the first things the host locks, together with the competition and ceremony venues, in the 24–12 month window before the IOI. The Accommodation Coordinator is therefore identified and active from the start of that window, with preliminary hotel contracts (or written options) in place long before the rest of the operational team is recruited.

Key responsibilities fall into three strands. The first is commercial: negotiating and managing the hotel contracts. The second is allocation: assigning rooms per delegation, keeping each delegation together and honouring partnered-participant data from registration. The third is on-site: acting as the single point of contact with hotel management during the IOI week, and coordinating check-in and check-out logistics with the Arrivals and Departure Coordinators.

When assigning rooms, contestants of opposite gender must not share a room. Similarly, leaders of opposite gender are not asked to share rooms, though they may explicitly request a shared arrangement. These rooming conventions follow host-country law and the norms of the participating delegations, and are therefore stated in binary terms; facility planning elsewhere in this document (for example the contest-hall toilet plan in Section 8.1) is instead written to accommodate every participating gender.

Every participant must have their own bed. Double, triple, and quadruple rooms are all acceptable and have been used at previous IOIs, but shared beds are definitely not recommended: each occupant should sleep in a single bed of their own. This needs to be checked explicitly when contracts are signed, because many hotels default to double beds in double rooms.

Hotel geography should be planned separately for contestants and leaders. Contestants should all be in one place: a single hotel, or at most a tight cluster within walking distance, so that the guide corps, medical cover, and daily briefings can be run as one operation and the quarantine separation from leaders is easy to maintain. Leaders tolerate more leeway and can be split across a small number of hotels if the contestant inventory requires it, but even here the spread must be modest; leaders travel to the same GA room, translation nights, and ceremonies as everyone else, and hotels scattered across a city multiply the transport load and fray communications.

If the chosen hotels or hostels do not provide towels or other standard amenities, delegations must be informed at least six weeks before the IOI.

If sponsors are organising activities or placing materials at the hotels (sponsor booths in hotel lobbies, or sponsor-hosted evening events), the Accommodation Coordinator must ensure the Branding Coordinator is informed and that proper coordination is in place to receive, store, and distribute sponsor goods and materials. The hotel must be briefed in advance about any sponsor deliveries or setup activities.

Key milestones:

- 24 months out: Hotel inventory secured (preliminary contracts or written options), together with the venues.

- 12 months out: Hotel contracts finalised and signed, in time to announce the host, location, and venue at the previous IOI.
- 2 months out: Room allocation plan drafted.
- 1 week out: Room assignments finalised from registration data.

15. After the Event

The host's work does not end on Day 8. This section covers the three post-event tracks that matter for the IOI community: debrief and handover, sponsor reporting, and archive. The HOC Chair should ensure each has a named owner before the week begins. Host-country financial and legal close-out sits alongside these and is owned by the operational budget owner identified in Section 5; it is a host-internal concern and is not covered further here.

Debrief and handover. Within two weeks of the IOI, while memory is fresh, the core team should hold a structured debrief capturing what worked and what did not. The resulting notes, together with running orders, supplier contacts, templates (registration exports, guide briefings, signage design files), and the material produced in preparation for this edition, form the handover to the next host. A single handover meeting, accompanied by a shared repository, is far more valuable than a pile of files without context. The formal handover to the IC takes place at the Winter Meeting, the next IC meeting after the IOI, where the outgoing host is expected to deliver a written report accompanied by an oral presentation on the lessons learned from the past IOI.

Sponsor reporting. Each sponsor contract specifies deliverables: logo visibility, social media reach, photographic rights, attendance at ceremonies. The Branding Coordinator and the Press & Media Lead assemble the evidence (photos of banners in place, social media analytics, press coverage, delivered goods) and report back within a few weeks of the event. Long-term IOI-wide sponsors are reported to via the IOI Treasurer; the host reports directly to its own host-specific sponsors. Keeping sponsors happy after the event is what makes them willing to support the next one.

Archive. Photos, video, printed materials, templates, the risk register, and the debrief notes should be archived at a stable location that survives the end of the host's organisational structure. A foundation, university, or national informatics body is usually the natural home. The archive should be indexed well enough that the next host can find a specific document, not only confirm that it exists somewhere.

16. Concluding Remarks

This document runs to more than twenty coordinator roles. The ideas behind them run through the whole text already: keep a backup for every key person, give each task one clear owner, make neighbouring teams talk to each other. I will not line them up again here. What I want to do in this last chapter is different. I want to set down the things the rest of the document deliberately leaves out, and the things I have learned only by watching IOIs from the inside.

What is not covered: I left out three areas on purpose, because they sit next to the operational plan rather than inside it.

One is the internal work of the HSC and HTC: how tasks are chosen and prepared, how graders are built, how the contest infrastructure is designed and tested. The IOI Regulations (IOI, 2021) cover that. Another is the long-term sponsor relationships that carry from one edition to the next; those are the IOI Treasurer's job, and I mention them here only where they touch an operational role. The last is the political and diplomatic layer (visa facilitation, government-level funding, host-country fundraising), which sits with the Host Steering Committee and the host's own sponsorship people.

What everybody underestimates: The IOI moves to a new country almost every year, and few countries host it twice, so nearly every host is doing this for the first time. I have watched many editions from the inside, and the same few things catch hosts out almost every time. The first is arrivals day, which is always longer and messier than anyone plans for. Every year a few teams turn up incomplete: an overbooked flight or a missed connection leaves a contestant or a leader stranded mid-journey, and someone on the arrivals team has to chase down where they are and when they will land. The second is catering: people eat more, and differently, than you expect. Not every meat eater likes every kind of meat, so they go looking for the vegetarian dishes instead, and the careful counts you prepared stop matching what actually ends up on people's plates. None of this is hard in itself. It is hard because you cannot see it coming from outside; it only becomes real once the event is running.

What should be shared between hosts: Every host builds the same things again from nothing: guide briefings, signage, bus schedules, room-allocation spreadsheets. Every host also delivers a debrief afterwards, but in practice it is taken in only by the next host and then goes no further. That knowledge should outlast a single handover, and that is what this document is for.

No two IOIs are alike, so do not copy the org chart in this document.

Drop roles, merge them, rename them, split them across more or fewer people: you have to do what makes sense given your venues, budget, team and other constraints. The one part I would keep is the check underneath it: go through every responsibility named here and make sure a real person has said yes to owning it, and make sure your coordinators understand enough of the rest of the event to catch the problems that fall between roles.

Acknowledgements

I need to thank the many members of the IOI community who proofread drafts of this article and whose suggestions shaped it into its current form. More than that, I want to thank them for the many wonderful evenings we have spent together over the years, talking about the IOI, sometimes late into the night. A great deal of this document began in those conversations.

References

- Matkarimov, B. T., Lee, G. C., Phillipps, M. & Schrijvers, E. (2017). IOI host guidelines: general aspects. *Olympiads in Informatics*, 11:175–192.
- Yusubov, A., Ahmadli, F. & Hasanov, J. (2022). Hosting IOI 2019 Azerbaijan: back to the future. *Olympiads in Informatics*, 16:173–195.
- International Olympiad in Informatics (IOI, 2021). IOI Regulations. Available at <https://ioinformatics.org/page/regulations/18>.
- International Olympiad in Informatics (IOI, 2026a). IOI Code of Conduct. Available at <https://ioinformatics.org/page/code-of-conduct/50>.
- International Olympiad in Informatics (IOI, 2026b). IOI Allies Programme. Available at <https://ioinformatics.org/page/ioi-allies/57>.

A Abbreviations

Abbreviation	Full form
A/V	Audio/Visual
CMS	Contest Management System
DSA	Distinguished Service Award
EGOI	European Girls' Olympiad in Informatics
GA	General Assembly
HOC	Host Organising Committee
HSC	Host Scientific Committee
HTC	Host Technical Committee
HVAC	Heating, Ventilation, and Air Conditioning
IC	International Committee (of the IOI)
IOI	International Olympiad in Informatics
ISC	International Scientific Committee
ITC	International Technical Committee

About the Author

Eljakim Schrijvers, though most people just call him Kim (the “elja” is silent, apparently), is deeply involved in the IOI community. He has been part of every EGOI since its start in 2021 and was the organiser of EGOI 2024 in the Netherlands.

He also contributes behind the scenes, having served for many years as Treasurer of the IOI. On the technical side, he runs contest systems for multiple Bebras and Kangourou Sans Frontières (KSF) countries, and has built the registration database used by many contests.

Kim also runs a YouTube channel where he explains algorithms in short, accessible videos, sometimes with more enthusiasm (and content) than production quality.

When he’s not working on his company, organising contests, or talking about algorithms, he can occasionally be found on a golf course, though not very successfully, by his own admission. He insists he enjoys it anyway.

About Journal and Instructions to Authors

OLYMPIADS IN INFORMATICS is a peer-reviewed scholarly journal that provides an international forum for presenting research and developments in the specific scope of teaching and learning informatics through olympiads and other competitions. The journal is focused on the research and practice of professionals who are working in the field of teaching informatics to talented student. OLYMPIADS IN INFORMATICS is published annually (in the summer).

The format for the journal follows the tracks:

- the primary section of the journal focuses on research
- the second report section is devoted to sharing experiences of countries in informatics olympiads
- the last smallest section presents books reviews or other information

The journal is closely connected to the scientific conference annually organized during the International Olympiad in Informatics (IOI).

Abstracting/Indexing

OLYMPIADS IN INFORMATICS is abstracted/indexed by:

- Cabell Publishing
- Central and Eastern European Online Library (CEEOL)
- EBSCO
- Educational Research Abstracts (ERA)
- ERIC
- InfoBase Index
- INSPEC
- SCOPUS – Elsevier Bibliographic Databases

Submission of Manuscripts

All research papers submitted for publication in this journal must contain original unpublished work and must not have been submitted for publication elsewhere. Any manuscript which does not conform to the requirements will be returned. The journal language is English. No formal limit is placed on the length of a paper, but the editors may recommend the shortening of a long paper. Each paper submitted for the journal should be prepared according to the following structure:

- concise and informative title
- full names and affiliations of all authors, including e-mail addresses
- informative abstract of 70–150 words
- list of relevant keywords

- full text of the paper
- list of references
- biographic information about the author(s) including photography

All illustrations should be numbered consecutively and supplied with captions. They must fit on a 124 × 194 mm sheet of paper, including the title.

The references cited in the text should be indicated in brackets:

- for one author – (Johnson, 1999)
- for two authors – (Johnson and Peterson, 2002)
- for three or more authors – (Johnson et al., 2002)
- the page number can be indicated as (Hubwieser, 2001, p. 25)

The list of references should be presented at the end of the paper in alphabetic order. Papers by the same author(s) in the same year should be distinguished by the letters a, b, etc. Only Latin characters should be used in references.

Please adhere closely to the following format in the list of references:

For books:

Hubwieser, P. (2001). *Didaktik der Informatik*. Springer-Verlag, Berlin.

Schwartz, J.E., Beichner, R.J. (1999). *Essentials of Educational Technology*. Allyn and Bacon, Boston.

For contribution to collective works:

Batissta, M.T., Clements, D.H. (2000). Mathematics curriculum development as a scientific endeavor. In: Kelly, A.E., Lesh, R.A. (Eds.), *Handbook of Research Design in Mathematics and Science Education*. Lawrence Erlbaum Associates Pub., London, 737–760.

Plomp, T., Reinen, I.J. (1996). Computer literacy. In: Plomp, T., Ely, A.D. (Eds.), *International Encyclopedia for Educational Technology*. Pergamon Press, London, 626–630.

For journal papers:

McCormick, R. (1992). Curriculum development and new information technology. *Journal of Information Technology for Teacher Education*, 1(1), 23–49. <http://rice.edn.deakin.edu.au/archives/JITTE/j113.htm>

Burton, B.A. (2010). Encouraging algorithmic thinking without a computer. *Olympiads in Informatics*, 4, 3–14.

For documents on Internet:

IOI (2008). *International Olympiads in Informatics*. <http://www.IOInformatics.org/>

Hassinen, P., Elomaa, J., Ronkko, J., Halme, J., Hodju, P. (1999). *Neural Networks Tool – Nenet (Version 1.1)*. <http://koti.mbnet.fi/~phodju/nenet/Nenet/General.html>

Authors must submit electronic versions of manuscripts in PDF to the editors. The manuscripts should conform all the requirements above.

If a paper is accepted for publication, the authors will be asked for a computerprocessed-text of the final version of the paper, supplemented with illustrations and tables, prepared as a Microsoft Word or LaTeX document. The illustrations are to be presented in TIF, WMF, BMP, PCX or PNG formats (the resolution of point graphics pictures is 300 dots per inch).

Contacts for communication

Valentina Dagienė
Vilnius University
Akademijos str. 4, LT-08663 Vilnius, Lithuania
E-mail: valentina.dagiene@mif.vu.lt
Website: <https://ioinformatics.org/page/ioi-journal>

Publisher office:

Vilnius University, Akademijos str. 4, LT-08663 Vilnius, Lithuania

June, 2026