

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" (blue) and "Watch Smart Strategy" (red). 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

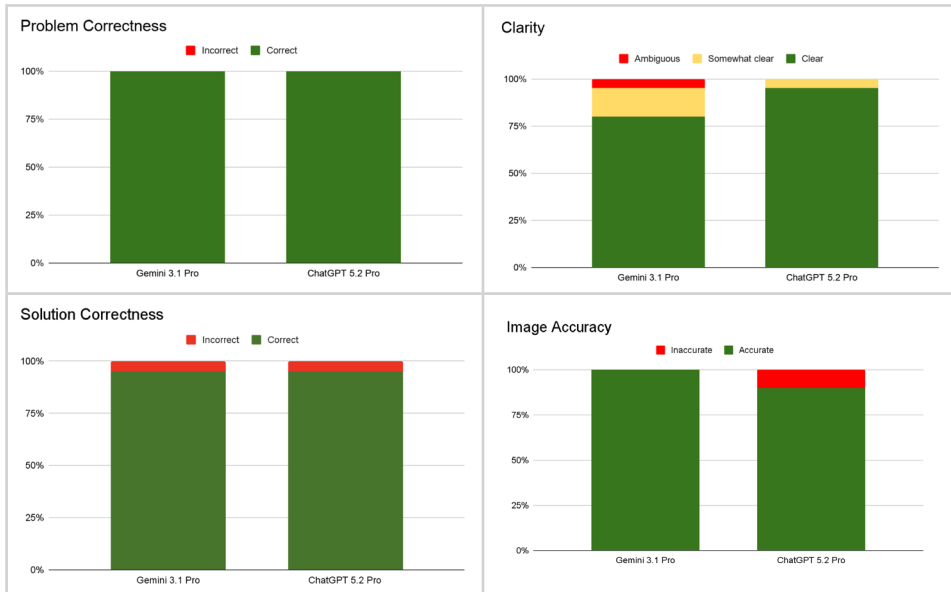
Table 3. ChatGPT 5.2 Pro

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.



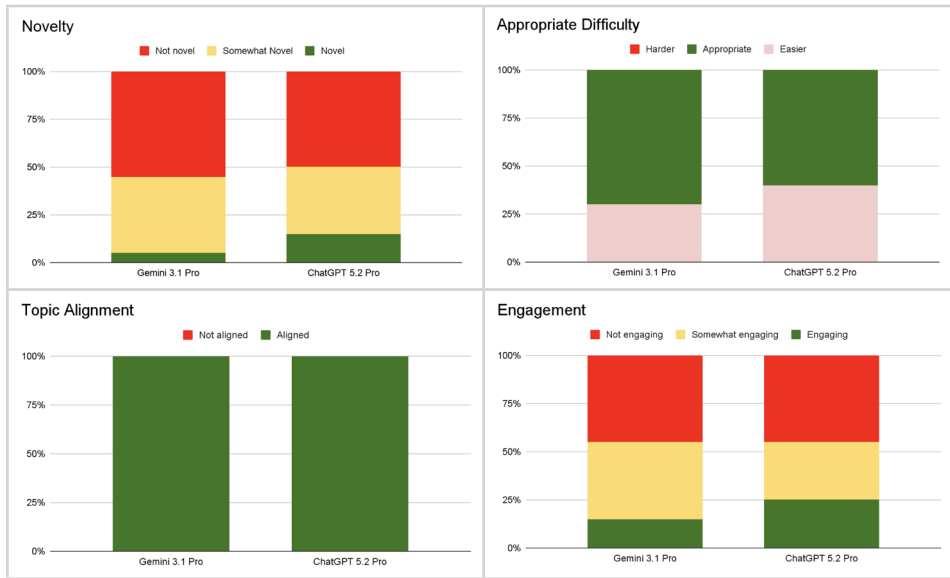


Figure 3. The distribution of scores for a single evaluation criterion

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.

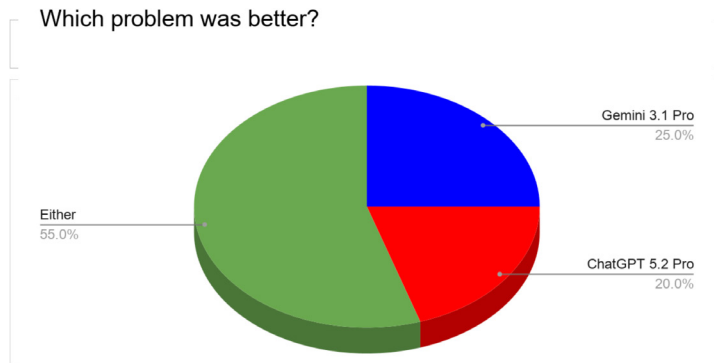


Figure 4. The experts' overall preference for the generated problems

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.