Olympiads in Informatics, 2017, Vol. 11, 3–18

DOI: 10.15388/ioi.2017.01

Constructive Problems in the Structure of the Olympiad in Discrete Mathematics and Theoretical Informatics

Vasiliy A. AKIMUSHKIN[1], Sergei N. POZDNIAKOV[2], Anton S. CHUKHNOV[3]

[1]*Informational and methodological center of computer technologies and informatics faculty of SPbETU "LETI"*
197376, Saint-Petersburg, prof. Popovs str., 5
tel. (812) 234-63-81, 26-96-493
[2]*Department of Mathematics, Saint Petersburg Electrotechnical University*
Mathematics and Mechanics Faculty, St. Petersburg State University
192281, Saint-Petersburg, Kupchinskaya str., 10-3-223
tel. (812) 234-63-81, (911) 76-98-492
[3]*Department of Mathematics, Saint Petersburg Electrotechnical University "LETI"*
194292, Saint-Petersburg, Rudneva str., 31/29 - 45
tel. (812) 234-63-81, (921) 976-96-87
e-mail: vasiliy.akimushkin@gmail.com, pozdnkov@gmail.com, septembreange@gmail.com

Abstract. In the paper problems which organizers of Olympiads are faced considered. The approach to solve this problems suggested. This approach is based on activity theory and includes using rather simple constructive problems as a first step to more complicated theoretical ones. The experience of implementing this approach within the framework of the Olympiad in Discrete Mathematics and Theoretical Informatics is described. The focus is set on computer manipulators – interactive dynamic models of mathematical and informatical object.

Keywords: Olympiads in informatics, discrete mathematics, computer science education, constructive problems, information technologies in education.

1. Introduction

One of the challenges of mass participation of schoolchildren in the Olympiads in mathematics and computer science is the complexity of the Olympiad problems. Most schoolchildren are not initially motivated to solve complex problems, so the number of participants in the Olympiads is rather small. At the same time, reducing the level of complexity of the Olympiad problems would undermine the very idea of the Olympiad movement, within the framework of which the student is offered difficult tasks that require a non-standard approach and a deep understanding of the problem. At first glance it seems that the contradiction between the difficulty of tasks and the mass participation of schoolchildren is insurmountable.

The approach, suggested in this article, showed the possibility of increasing the number of participants of the Olympiad by using of constructive tasks. Those tasks are tied to computer dynamic models – manipulators dealing with objects of the problem. It allows to bring an experimental component into the solution of theoretical problems, to "touch" the idea of a solution "by hands", to transform it into a constructive form and only after that to make theoretical generalizations. Built on computer manipulators, representing opportunities for experiments with important theoretical concepts, such problems play the role of a bridge to more complex theoretical problems (Akimushkin and Pozdniakov, 2015).

Since the Olympiads are usually held in several stages (for example, training, qualifying and main stages), it is proposed to consider all stages as components of a single process of leading schoolchildren to difficult theoretical problems; to ensure continuity of the Olympiad stages, in order to introduce new theoretical concepts into the subject of the Olympiad; to begin with simple constructive tasks on the training tour and gradually complicate them, adding theoretical questions to the qualifying round and complete with serious theoretical tasks in the final round.

In this way it is possible to solve one more problem – the problem of introducing new ideas into existing mathematics and computer science courses. At the Olympiad, the topic of the tasks may be wider than the one presented in the school curriculum, but new ideas should be introduced the way that they can be understood by schoolchildren. Three-step introduction of new concepts through the use of constructive tasks on computer manipulators allows us to solve this problem.

2. Background and Literature Review

The approach, suggested in the article, based on the activity theory. Psychological works of L.S. Vygotsky about the role of a tool in the environment of a child state and justify that men and women obtain the control of their own intelligence only through the control of the real world objects. In what follows the names of these objects in a language used by men and women become the signs of these objects, and thus the language is inseparable from their thinking (Vygotsky, 1930);

The works of Seymour Papert actually used Vygotsky ideas in the conditions of the appearance of a computer and "smart objects" in a child environment. They are introduced into a child environment and through the interaction with them it forms his representation of important scientific ideas and analyzes its own thinking (Papert, 1980);

For the last 10–15 years, there has been rapidly growing the movement of mass competetions, which tend to occupy an intermediate place between the school course of computer science and mathematics and the Olympiad movement (van der Vegt, 2016; Kostadinov *et al.*, 2015; Sysło and Kwiatkowska, 2015). The most famous contest is the BEBRAS competition, which brought together the methodological ideas of scientists and practitioners from more than 60 countries in overcoming this challenge (Dagienė and Sentance, 2016). Starting from multiple choice test problems, which connects the form of the contest with various systems of knowledge assessment, over time the competition absorbed the ideas of other researchers and the range of possible types of answers in tasks was expanded by so-called dynamic tasks.

Dutch Olympiad of Informatics can be an example of using constructive and theoretical non-programming tasks in the framework of informatics Olympiad with positive effect (van der Vegt, 2012; van der Vegt, 2016).

Another example of an informatics competition using theoretical problems is the Open School Olympiad "Information Technologies" held by IFMO University in Saint-Petersburg.

The team represented by the authors of the article has also been working on the idea of introducing electronic tools into the process of solving problems in mathematics and computer science for more than 15 years, and during this time supports the CTE contest ("Konstruiruy, Issleduy, Optimiziruy" which means CTE "Construct, Test, Explore"), based on the idea of using computer models of meaningful ideas from mathematics, physics and informatics for cultivation of interest in science through experimental and constructive activity (Ivanov, *et al.*, 2004; Posov and Maytarattanakhon, 2014).

3. Olympiad Structure

Olympiad of the Olympiad in Discrete Mathematics and Theoretical Informatics consists of three stages: training, qualifying and final.

Training round serves to let participants get acquainted with the framework, with computer manipulators and with basic concepts of discrete mathematics and theoretical informatics used in the tasks. It is held on the internet, the participants are able to perform the tasks at home. Tasks of the training round are rather simple and score gained by the participants has no consequences for them. Tasks are checked automatically.

Qualifying round is open and is also held on the internet. Tasks are harder then in the training round and require some thinking. Most of the tasks use computer manipulators, they are checked automatically as in the training round. A few tasks do not require manipulators, but text solution. They are checked manually.

Russian participants of the BEBRAS competition which takes place just before the start of the training tour of the DM&TI Olympiad are invited to the Olympiad,. Through this, the number of regions of Russia in which the olympics is held is rather big. For example in the season 2016–2017 among the participants of the qualifying round were representatives of 60 regions of Russia (out of a total of 85).

The final round is organized on the sites of universities of those cities, which are convenient for participants who have taken part in this round. In 2016–2017, 18 sites were organized for the final round.

The final round takes place on the server of the Olympiad in the same environment as the training and selection rounds. Thus, all participants are already familiar with the shell, and the local co-organizers need only identify the attendees and control the independence of their work. The logging of works is carried out centrally.

Tasks of the round belong to both manipulator and theoretical types. As the number of participants is lower than in qualifying round so there is a room to increase number of tasks of the second type.

After a qualifying as well as final round the surveys were held. Questions and resuls of both surveys are presented in the «Results analysis» section.

In the table below we can see a number of participants of qualifying and final rounds as well as number of participants completed the surveys.

Round	Participants	Surveys completed	%
Qualifying	513	127	25%
Final	97	35	36%

4. Computer Manipulators as a Basis for Constructive Tasks and Experiments

We call computer manipulator a dynamic interactive model of object of greatest importance for discrete mathematics and theoretical computer science. Each manipulator gives participants enough freedom in changing parameters of a task. Thus we allow participants to get closer to objects of discrete mathematics and theoretical informatics, to make them master concepts and methods not only through reading their formal definitions, but as well through their own practical experience, throgh «feeling» and «touching» properties of those objects. This process, which could be called interiorization, allows theoretical ideas to find their way into the student's minds.

In the 2016–2017 Olympiad the following manipulators were used:

- **Graphs** – a manipulator based on the ability to build a graph, introducing the necessary notation, edge weights, coloring vertices, etc. The manipulator may be adapted to the certain task using only those instrumental capabilities that are required to complete this task to make the interface intuitively understandable and not overloaded with unnecessary operations. Also, the manipulator is equipped with a set of procedures for verifying the properties of the solution that make up the condition of the task (for example, checking the planarity or regularity of the graph being built) that the participant of the Olympiad can use for experiments in the process of solving constructive problems or investigating particular cases of the theoretical problem (Akimushkin *et al.*, 2015). These properties belongs to all the manipulators used in the Olympiad.
- **Finite state machines** – the manipulator showing the graphical representation of the finite state machine. It can be used in different contexts, for example, as a deterministic machine, as an machine with an output alphabet or a recognizing machine. Just as in the "Graphs" manipulator, the machine used in the certain task has already been adapted to the task, and the user works within the context of the task restricted by built-in constraints of the manipulator.

- **Regular expressions** – the manipulator is another form of representation of content, which, according to Kleene's theorem, can be formulated in terms of finite state machines. However, the presence of several representations of the same concept is a way of supporting the process of assimilation of this concept (Bogdanov *et al.*, 2008). Despite the fact that the solutions of constructive problems on this manipulator, as well as on the Finite State Machines manipulator, can be verified algorithmically, in both cases the estimation of particular solutions with respect to a parameter is used-the percentage of properly analyzed or generated chains.
- **The Turing Machine** – this well-known manipulator allows you to introduce algorithmic elements inherent in programming Olympiads into the Olympiad in theoretical informatics. At the same time, since the Turing machine is a theoretical tool for investigating the algorithms, there appears the natural possibility to raise questions of the existence or laboriousness of algorithms with given properties.
- **Logical schemes.** While the above manipulators are very slightly connected with the ideas of the school curriculum, this manipulator is very close to the elements of logics included in the course of informatics in Russia (proposition logic). Represented by logical schemes, Boolean functions (logical expressions) are more demonstrative and easier to set up constructive tasks with. The user can easily test the assembled circuit on all binary sets (also the percentage of sets on which the circuit works correctly, is used to evaluate partial solutions). In addition, schemes allow to offer tasks that go beyond the elements of mathematical logic studied in school and are related to real problems of constructing logical elements that can have several outputs and some parts of the circuit can use to calculate signals at several outputs (that is, the task is not reduced to the calculation of the values of several independent Boolean functions).
- **Tarski world –** this is the most complex manipulator that allows students to enter the field of first-order logic or the logic of predicates. The manipulator is based on the idea first implementatied in the well-known program with the same name (Barker-Plummer, *et al.*, 2008) and was used earlier in the CTE competition. In this manipulator, in an informal (verbal) form, you can use both logical operators and quantifiers. As a basic subject set, a checkered field with the figures placed on it is used. Figures have properties such as color, shape, size (single predicates) and a certain location relative to each other, for example, higher, lefter, side by side (two-place predicates).

5. DM&TI Olympiad Task Analysis

As already mentioned above, tasks are formulated in a way to create cross-cutting thematic lines through all stages of the Olympiad, except that in each round the task authors try to create internal links between tasks so that participants in a short time of the Olympiads do not scatter attention to different subjects, but, on the contrary, to see one story from different sides, in different interpretations and metaphors. Therefore, the problems of one stage, usually about 8, are generally divided into 3–4 groups of problems so that the tasks of each group have some unity.

Let us give an example of several such interrelated tasks from the qualifying round of the Olympiad in Discrete Mathematics and Theoretical Informatics of the 2016–2017 season (DM&TI-2017).

Let's consider a constructive task on regular expressions (the original task number is given) and the manipulator image with which the participant works.

№2 (3 points)

*Construct a regular expression describing the set of words from the letters **a** and **b**, from which all words specified by the regular expression **(ab)*** are removed. Try to give the expression as short as possible.*

Help (can be called by the participant).

Regular expressions contain three operations: splicing strings (multiplication), selecting one of the two options (addition) and an iteration, denoted by an asterisk. The initial solution is **b*(a + b)**. It consists of two parts – **b*** denotes an arbitrary number of letters **b** (possibly none), **(a + b)** – one of the letters **a** or **b**. Below through the color highlighting you can see which words do satisfy this expression, and which do not.

Solution

If the word does not satisfy the regular expression **(ab)***, i. e. does not have the form **abab ... ab**, it means that either this word starts with **b**, or ends with **a**, or contains two identical letters in a row. The first term in the formula corresponds to the first case, the second term to the second case, the third one to the third.

The answer is shown on the Fig. 1:

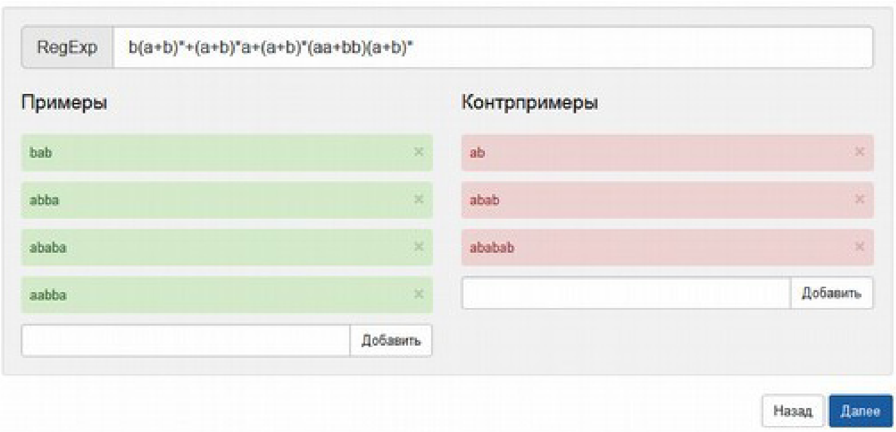

Fig. 1. User interface for working with constructive tasks on regular expressions. A. participant can extend a number of examples and counterexamples to check the input of the response (the top line shows the answer of the problem).

Immediately after this task is the combinatorical problem, based on the same interpretation – representing certain set of words in the form of regular expression.

№3 (4 points)

How many different words does the regular expression ***(a + ab) (b + ab) (a + ab) (b + ab) (a + ab)****? Do not forget to explain your answer.*

Solution

Each expression in braces can have one of two possible values, which means that there are 32 variants in all. However, some words are thus constructed two times. Only two words are given in two ways, the others by one. Thus, the answer for this problem is 32 – 2 = 30.

The next problem does not differ from Problem 2 in essence, but is proposed in a different interpretation – the interpretation of finite state machine. Obviously, the participant will not fail to notice the same condition in these tasks. Thus, he will be indirectly get acquainted with the concepts necessary for the formulation of Kleene's theorem on the equivalence of regular expressions and automaton languages.

№4 (4 points)

Below there is a finite state machine recognizing all words from the letters ***a*** *and* ***b*** *that match the regular expression* ***(ab)*****. Rework it to make it recognize all the words in the same alphabet, except these. Try to keep the machine as small as possible.*

Solution:

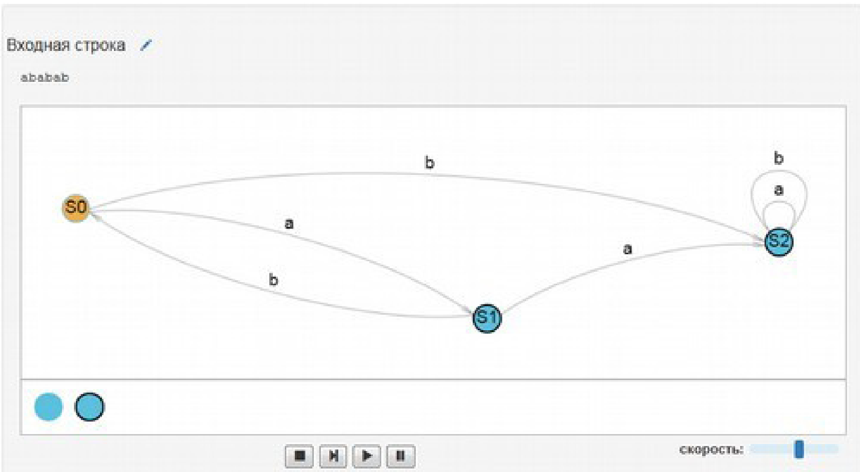

Fig. 2. User interface for working with finite state machines constructive tasks (the figure shows the solution of the task).

Now let us consider the tasks of the final round. In the 2016–2017 season, some of the tasks were gathered with the concept of the "voting machine" (note that this task was used in the popular interpretation in the CTE competition). Here are the tasks texts and some solutions.

The first series of problems is based on the manipulator "Logical schemes" and represents an alternation of constructive tasks and theoretical problems.

In the core of all tasks there lies a fact that goes beyond the school curriculum. From the school curriculum schoolchildren can learn the fact implicitly presented in it about the completeness of the set {negation, disjunction, conjunction} (or even with the exception of a disjunction or a conjunction according to Morgan's law). However, in these problems we consider a self-dual, monotonic Boolean function that preserves 0 and 1, which must be expressed in terms of monotone functions preserving 0 and 1. The last two problems represent a significant complication of the original problem, since in this case the function through which the function given in addition to the listed properties also possesses self-duality.

1. Logical schemes: "Voting machine"

*Let's denote a **voting machine** for an odd number **n** a logic scheme with **n** inputs that return «true» when more than half of the inputs have «true» values, and returns «false» value otherwise.*

*This is a very natural definition: if **n** inputs are **n** people, each of which votes either «for» (true) or «against» (false), on the output we get the option for which the majority has voted.*

1.1. (3 points)

Construct a voting machine for three people using the AND and OR logical elements.

Solution:

Possible solutions are shown below:

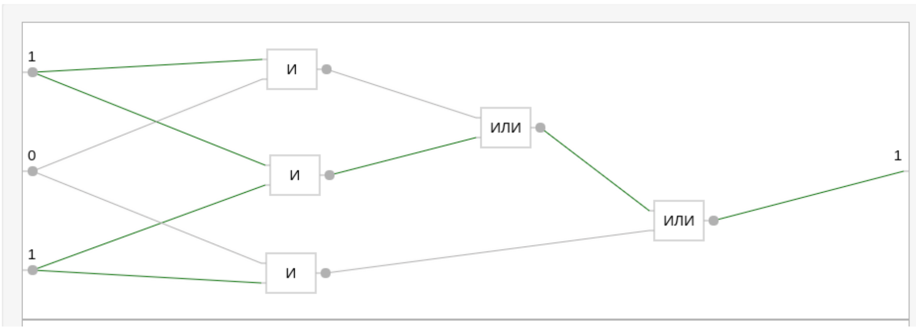

Fig. 3. User interface for working with constructive tasks on logic circuits. One solution for the task.

Also on the diagram you can see how this logical scheme works on a certain set of values (TRUE, FALSE, TRUE).

At the Fig. 3 we sort all possible pairs of inputs, and if at least one of them takes a true value, the logical scheme produces the true result.

1.2. (3 points)   *Prove that it is possible to construct an automaton for voting of any odd number of elements from the AND and OR elements.*
1.3. (4 points)   *Prove that 2^n elements AND and OR are sufficient to construct a voting machine for odd number n people.*
1.4. (4 points)   *There is a logical element with three inputs, which gives output «1» if the number of «1» at the inputs is greater than the number of «0» (implements the voting function for three people).*   *Construct a voting machine for five people, using only voting elements for three people.*
Solution: We enumerate the voters using numbers from 1 to 5. Let's take the voting elements of three people with numbers 1, 2 and 3; 1, 2 and 4; 1, 2 and 5. The outputs of each of these elements we submit to the inputs of the fourth voting element for three people. Notice that the scheme obtained gives the correct result for voting of five people for all situations except two: when people with numbers 1 and 2 vote «for», and the other three «against», and the inverse situation.   We will create two more similar schemes: a scheme that is «mistaken» only in the distribution of the voters 2 and 3 against 1, 4 and 5, and a scheme that «mistakes» in the distribution of 4 and 5 against 1, 2 and 3. Note that at least two of these schemes will give the correct result for any distribution of votes. So, if we submit their outputs to the inputs of the final voting element for three people, the result will always be correct.

Let us pay attention to the solution of the previous problem (the corresponding scheme can be easily constructed and is not given in the article). This constructive problem leads us to the solution to the next, theoretical problem. This allows us to make an assertion that has long time been accepted by mathematicians and is confirmed by psychologists that theoretical knowledge is based on very certain tasks, solved by the person himself.

1.5. (5 points)   *There is a logical element with three inputs, which gives output «1» if the number of «1» at the inputs is greater than the number of «0» (implements the voting function for three people).*   *Prove that it is possible to build a voting machine for any odd number of people, using only voting elements for three people.*

Solution:

We prove this statement using mathematical induction. The basis n = 3 is contained in the problem text.

To make induction step form 2n −1 to 2n + 1 we use the inductional hypothesis. Regard the following voting elements for 2n – 1 people: all but 1st and 2nd; all but 2nd and 3rd; all but 1st and 3rd. Their outputs submit to the inputs of the element for three people.

The scheme obtained will give us correct results in almost all cases. The only exception is the situation when majority contains exactly n+1 people including 1st, 2nd and 3rd.

Creating another two schemes which are «mistaken» in two another situations. Then when we submit outputs of all 3 schemes to the input of the final three people element, the result provided by this element will be always correct.

In tasks of other manipulators, the idea of modeling the voting scheme was also used. So, for problems on regular expressions and finite state machines, it was suggested to describe the structure of input data, in which voting ends with a positive result.

3.1. (3 points)

There are N people standing in the queue for voting. It is known that next to each person (directly ahead of him in line or behind) is a person who votes «for». Prove that the number of people in the queue, who vote «for» at least half of the total amount of people.

3.2. (3 points)

We assign to each person in the queue «1» or «0», depending on whether he votes «for» or «against». It is known that next to each person (directly ahead of him in line or behind) is a person who votes «for». Construct a regular expression that describes all such sets of 0 and 1 or prove that this is impossible.

4.1. (3 points)

We assign to each person in the queue «1» or «0», depending on whether he votes «for» or «against». It is known that next to each person (directly ahead of him in line or behind) is a person who votes «for».. Construct a finite state machine that recognizes all such sets of «0» and «1» or prove that this is impossible.

6. Results Analysis

During the Olympiad we were evaluating the hypothesis: solving simple constructive tasks may be an important step to more complicated theoretical ones; theoretical tasks become avainle to be solved by more participants due to existance of constructive ones;

constructive problems are helpful participants not only in solving specific tasks but in deeper understanding of the whole subject as well.

Of course the optimal way to make an experiment is to check the theoretical tasks with and without constructive ones on different groups; unfortunately, it is completely impossible within the same competition. So the main thing we should analyze as a results of the experiment is participants feedback: whether constructive problems were crucial for them or not.

Constructive tasks are being solved by much more participants than theoretical tasks, at the same time, participants who have passed into the final round generally solve the theoretical tasks of the qualifying round.

6.1. *Qualifying Round Results Analysis*

The survey of participants of the qualifying round (127 people fulfilled the survey, which is about 20% of all participants in the qualifying round) shows that participants liked the tasks (96% of participants liked the participation in the Olympics of the DM&TI). This answer can be interpreted as the fact that the tasks did not cause rejection reactions, were understandable and "accepted" by the participants.

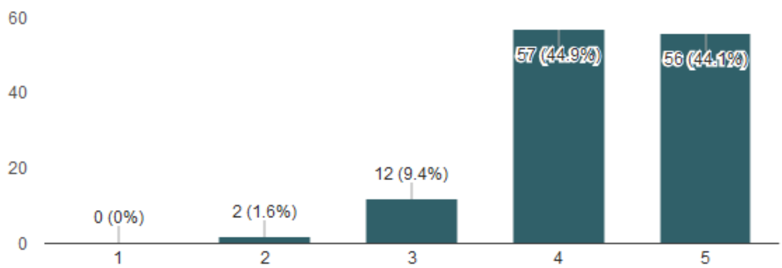

Fig. 4. Results of a survey of participants on how much they liked the tasks (on a five-point scale from 1 to 5).

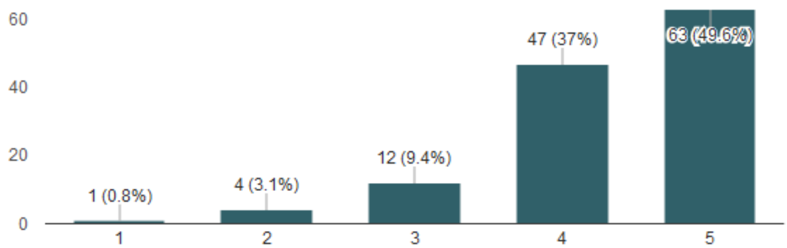

Fig. 5. Results of a survey of participants on how much they liked working with dynamic modules (on a five-point scale from 1 to 5).

It should be noted that the distribution of answers to the question, whether participants liked working with dynamic manipulator models, has a similar structure.

Some statements of the participants of the qualifying round:

> *«Most of all I liked interactive tasks, such as graphs, logic schemes, etc. I liked them because I could interact with the task «visually», without text and commands. Visual contact is very important.»*

> *«You could put the dynamic modules in permanent access, both for training and just as a useful application.»*

> *«It's quite interesting and unusual that you can use more opportunities of the online Olympiad (computer manipulators). The content of the Olympiad itself is also impressive, there were no special questions on the tasks or places where it takes more time to think about the problem than to solve it.»*

These data show that the problem of attracting more participants to participate in the qualifying round is solved by the use of dynamic modules and constructive tasks built on them.

6.2. *Final Round Results Analysis*

In the final round of the 2016–2017 season people participated (of 104 passed). 13 tasks, divided into 6 groups by reference to various dynamic manipulators were offered to participants. Initially, the number of tasks was assumed redundant, since for 3 hours of a final round it is really possible to solve 6–7 tasks. The total amount of score of all tasks

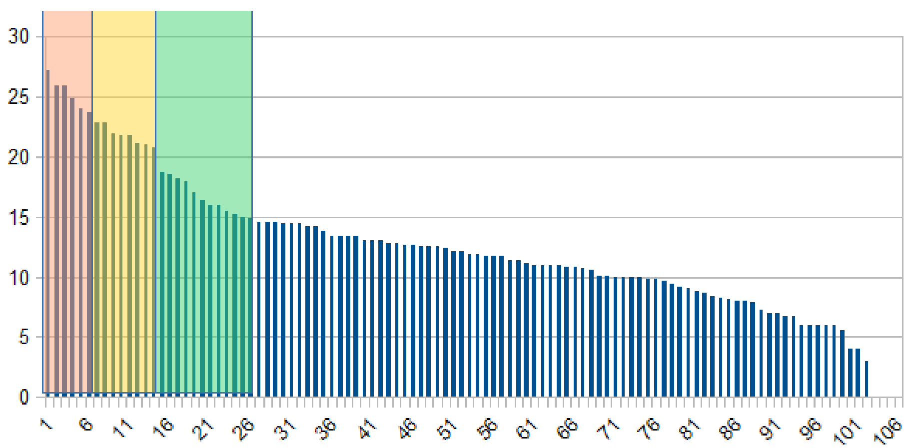

Fig 6. Score distribution of the final round. Diploma of different degrees areas are coloured: rose – I degree; yellow – II degree; green – III degree.

was 51 points. Half of these points is the expected best result. After the final round, the following criteria were determined: more than 23.5 – I degree diploma, 20 to 23.5 points – II degree diploma, 15 to 20 points – III degree diploma. The median score obtained is 25% of the maximum.

35 of 97 participants of the final round took part in the survey the targeted to determine the role of constructive problems in solving theoretical ones and the influence of identical manipulators and theoretical ideas attached to them throughout all rounds of the Olympiad over preparation for the final round. Results of the survey are presented in the figure 7.

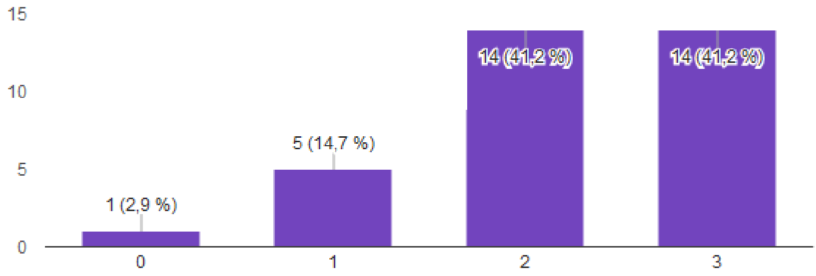

Fig. 7. Results of a survey for the participants of a final round about the importance of constructive tasks for understanding theoretical problems associated with them (on a four-point scale: 0 – do not influence, 3 – are necessary for solving theoretical problems).

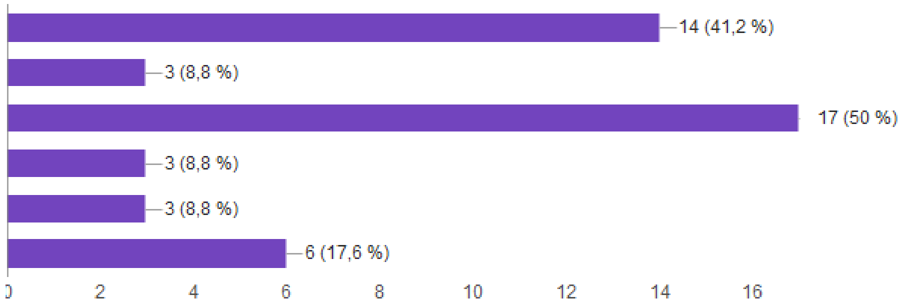

Fig. 8. Results of the survey of the final round participants about how difficult the problems of the Olympiad revealed to be:

line 1 – «these problems are difficult, because they are not in the school curriculum»
line 2 – «tasks are more difficult than in other Olympiad»
line 3 – «thanks to training and qualifying rounds I got acquainted with the new subjects of the problems and began to solve them successfully»
line 4 – «I am familiar with this kind of tasks through additional activity in the school»
line 5 – «I participated in this Olympiad more than once and got used to the specific tasks of the Olympiad»
line 6 – «other».

Statistical analysis shows that the confidence interval for the average estimation of the importance of constructive problems for the successful solution of the theoretical is (2.01, 2.39). With a probability of 0.95, it can be argued that the average value when sampling a larger volume will not go beyond the interval found. This means confirmation of the assumption that the constructive tasks that precede theoretical problems play an essential role for their successful solution.

Figure 8 shows the results of a survey about the difficulty of the tasks. The third line of the chart is highlighted most clearly in which the participants report on the role of the training and qualifying round for the preparation for a final round: «Thanks to the training and selection round, I got acquainted with the new subjects of the problems and began to solve them successfully».

The estimayed average per cent of participants, for whom the training and qualifying rounds became important elements of preparation for the successful completion of the final one, ranges from 43% to 57%. With a probability of 0.95, it can be argued that the average percent when sampling a larger volume will not go beyond the interval found.

7. Conclusions

Based on five years of experience in organizing the Olympiad in Discrete Mathematics and Theoretical Informatics and in analyzing the results of the Olympiads and questionnaires, the following conclusions can be drawn:

1. Using constructive tasks built on computer manipulator models allows to increase in the number of participants in the training and qualifying rounds of the Olympiad.
2. The absence of important concepts used in Olympiad problems in the school curriculum in mathematics and computer science can be compensated by considering three rounds of the Olympiad as a single process of immersing the participants of the Olympiad in new subject areas. The latter can be considered as a pre-process for introducing new elements into the school curriculum.
3. Using a series of tasks, the first ones of which have a constructive form and allow you to experiment with solutions of problems in a computer-modeled domain, allow participants to successfully solve theoretical problems, removing the barrier of fear of difficult tasks.

References

Akimushkin, V.A., Maytarttanakhon, A., Pozdnyakov, S.N. (2015). Olympiad in theoretical computer science and discrete mathematics. In: Brodnik, A., Vahrenhold, J. (Eds.) *Informatics in Schools: Curricula, Competences, and Competitions. 8th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives: ISSEP 2015, Ljubljana, Slovenia, September 28 – October 1, 2015, Proceedings*. Springer, LNCS 9378, 94–105.

Akimushkin, V.A., Pozdniakov, S. (2015). Computer enhanced Olympiad in theoretical informatics and discrete mathematics In: Vassiliev N.N. (Ed.), *International Conference Polynomial Computer Algebra 2015. St.-Petersburg, April 13–18*. Euler International Mathematical Institute, VVM. Publishing, 80–83.

Barker-Plummer, D., Barwise, J., Etchemendy, J. (2008). *Tarski's World*. Stanford, Calif: CSLI Publications.

Bogdanov, M., Pozdnyakov, S., Pukhov, A. (2008). Multiplicity of the knowledge representation forms as a base of using a computer for the studying of the discrete mathematics. In: *The 9th International Conference "Teaching Mathematics: Retrospective and Perspectives"*. Vilnius Pedagogical University, 16–17 May 2008.

Dagienė, V., Sentance, S. (2016, October). It's computational thinking! Bebras tasks in the curriculum. In: *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*. Springer International Publishing, 28–39.

Ivanov, S., Mamaeva, S., Pozdnyakov, S., Stepulenok, D., Entina, S. (2004). Computer support of educational research on mathematics. *Computer Tools in Education*, 2, 5–18. (In Russian. Иванов, С., Мамаева, С., Поздняков, С., Степулёнок, Д., Энтина, С. Компьютерная поддержка дистанционного учебного исследования по математике. *Компьютерные инструменты в образовании*, 2, 5–18).
`http://window.edu.ru/resource/340/24340/files/2004_2_05.pdf`

Kostadinov, B., Jovanov, M., Stankov, E., Mihova M., Risteska Stojkoska, B. (2015). Different approaches for making the initial selection of talented students in programming competitions, *Olympiads in informatics*, 9, 113–125. DOI: `http://dx.doi.org/10.15388/ioi.2015.09`

Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas.

Posov, I., Maytarattanakhon, A. (2014). Automation of distance contests based on research problems in mathematics and informatics. *Computer Tools in Education*, 6, 45–51. (In Russian. Посов, И.А., Майтараттанакон, А. Автоматизация проведения дистанционных соревнований, основанных на исследовательских сюжетах по математике и информатике. *Компьютерные инструменты в образовании,* 6, 45–51).

Sysło, M.M., Kwiatkowska, A.B. (2015). Introducing a new computer science curriculum for all school levels in Poland. *LNCS*, 9378, 141–154.

van der Vegt, W. (2012). Theoretical tasks on algorithms; two small examples. *Olympiads in Informatics*, 6, 212–217.

van der Vegt, W. (2016). Bridging the gap between Bebras and Olympiad; experiences from the Netherlands. *Olympiads in Informatics*, 10. 223–230.

Vygotsky, L., Luria, A. (1930). Tool and symbol in child development. In: Jaan Valsiner and Rene van der Veer (eds.), *The Vygotsky Reader*.

V. Akimushkin. Graduated from Saint-Petersburg State Electrotechnical University "LETI", Computer Science Faculty and then from Saint-Petersburg State Universtity. Working at Informational and methodological center of computer technologies and informatics faculty of SPbETU "LETI" as a leading programmer.

S. Pozdniakov. Graduated from Leningrad university, Faculty of Mathematics and Mechanics. Doctor of sciences in theory and methodology of mathematics and computer science 1999 (Moscow). Doctor thesis title: “The modelling of information environment as technological base for teaching mathematics”. Working at Saint-Petersburg State Electrotechnical University “LETI” from 1986 and at Saint-Petersburg State Universtity from 2007. Teaching courses for university students: “Discrete Mathematics”, “Mathematical Logic”, “Theory of Algorithms”, “Construction of Computer Programs for Education”, “Information Technologies in Mathematics”, “Models of Intellectual Processes”. Teaching courses for school teachers: “Methodology of Using Computer for Supporting of Math Education”.

A. Chukhnov. Graduated from Saint-Petersburg State Universtity, Faculty of Mathematics and Mechanics at 2004. Working at Saint-Petersburg State Electrotechnical University “LETI” since 2005, specialized in Discrete Mathematics and Mathematical Logics. Also since the same time working at school №239 with talented schoolchildren who are interested in additional education.