

ANALYSIS OF THE UNDERSTANDING OF THE MATERIAL OF THEORETICAL INFORMATICS IN COMPETITIONS AND OLYMPIADS IN INFORMATICS*

Pozdniakov S. N.¹, Chukhnov A. S.¹, Pangina N. N.¹

¹Saint-Petersburg Electrotechnical University, Saint Petersburg, Russia

²Municipal autonomous educational institution of additional education
“Center for Information Technologies”, Sosnoviy Bor, Russia

Abstract

Contests and Olympiads should be considered not only as a way of comparing the students' levels of training. They can be regarded as a different way of organizing the mental activity of participants. Different types of tasks initiate various types of intellectual activity among students. The Olympiad tasks differ from “school” ones by non-standardness, a combination of the understandability of the formulation and the non-obviousness of the solution, the possibility of various approaches to solving the problem.

At the same time, the variety of contests and Olympiads raises the task of investigating and classifying them, focusing on supporting different styles of thinking. In this paper, we carry out an analysis of contests and Olympiads in informatics, aiming at identifying aspects of the student's intellectual activity, which are not sufficiently evaluated at school.

The Olympiad in Discrete Mathematics and Theoretical Informatics is at the intersection of the two sciences. The article deals with the problems encountered by the organizers of competitions in mathematics and computer science and demonstrates some ways to solve them using the example of the Olympiad.

Keywords: *mathematical olympiads, programming olympiads, constructive tasks, constructive problems, distance learning, constructive activity, computer tools, subject manipulators.*

Citation: S. N. Pozdniakov, A. S. Chukhnov, and N. N. Pangina, “Analysis of the Understanding of the Material of Theoretical Informatics in Competitions and Olympiads in Informatics,” *Computer tools in education*, no. 2, pp. 55–67, 2018.

INTRODUCTION

After Unified State Exam became not only final exam for schoolchildren, but an entrance exam to Russian universities as well, Russian universities faced up with a question of how to ensure the quality of entrants and later students in this conditions. One of the main problems is that, according to current regulations, the result of the exam that should be accepted by a university, is a sum of three different marks with equal weight. For technical specialities the entrants have to pass exams in mathematics, Russian language and informatics or physics. It seems unfair that possibly talented student with good results in profile subjects could fail to

*The work was supported by the Russian Foundation for Basic Research (Project No 18-013-01130).

enter the desired university due to rather bad mark in Russian, which is probably a bit irrelevant to his or her future profession.

For these students there is another way to get to the universities. There are huge amount of different contests and Olympiads. For this paper Olympiads in mathematics and informatics matter. The main events of the year are All-Russian Olympiads of Schoolchildren in certain disciplines. Its winner is free to enter any university in our country. Besides that, there is every year published list of Olympiads approved. Their results should be recognized by any higher education institutions. In 2017/2018 season there are 97 different Olympiads in this list [1], including 26 in mathematics and 16 in informatics.

The main goal of the All-Russian Olympiads of schoolchildren is the identification of gifted students. Participation in an Olympiad for the student is an opportunity to get a more objective assessment of his personal qualities, the ability to relate himself among his peers, to realize himself, expand his horizons and his social circle, this is the hope to achieve high results, to satisfy his ambitious demands. Olympiads open up to the pupil numerous opportunities... especially in informatics and IT. The combination of enormous volitional, physical, moral and intellectual qualities is an unordinary personality... striving for new knowledge and new horizons [2].

Despite those Olympiads formally do not require any knowledges outside the school program, to be good at school is definitely not enough for successful participation. To achieve good results on the Olympiad, students should develop non-standard thinking and be available to combine different topics of the Olympiad subject, which they probably never do at school. The way to improve skills needed to succeed at the Olympiads are not only extracurricular studying. Olympiads themselves can not only be a reason for students to improve their skills and knowledges but also can serve a tool for it.

Since mathematics and computer science have a lot of intrasubject connections, in Russia they are united into one educational area (although they are taught as different subjects) [12], so the important task of the Olympiad movement in this educational area is to rely on these links.

This paper is related to the study of the connection between computational thinking [9, 10] and creative thinking [15]. There is regarded the influence of constructive tasks based on working with computer tools on solving Olympiad tasks associated with them. The work is a continuation of the study on the use of constructive problems at the Olympiads in Informatics [13].

1. ANALYSIS OF COMPETITIONS AND OLYMPIADS IN INFORMATICS AND MATHEMATICS IN TERMS OF WAYS TO ORGANIZE INTELLECTUAL ACTIVITY OF STUDENTS

Most of informatical Olympiads are actually programming contests focused on efficient algorithms with preset criteria of the efficiency. Submitting a solution in the form of computer program leaves to the participant a kind of independence in choosing the way of solving the problem. Also it allows assessing the solution without a direct interaction between the jury and the participant through the series of tests.

The negative side of this approach is that you never can evaluate aesthetic aspect of the solution, logics of the program, to appreciate well-organized thinking. The goal to solve the task is sometimes substituted by the goal to pass as much tests as possible, so brute force programs and programs which are written worse could be more efficient here than correct programs based on reasonable algorithms. Sometimes it could be a reason for some talented students to prefer project activity rather than Olympiad one.

International “Bebras” contest [5, 6] provides another view at the competitions in informatics. The tasks in this contest are multiple choice ones or, preferably, the “dynamic” constructive tasks where participants have to “touch” an informatical object and solve a problem of search or sorting or something else manually. Mathematics is considered to be excluded from the task as much as possible, with focusing on non-mathematical essences, such as data representation and algorithms, but of course can never be excluded at all.

At the CTE (Construct-Test-Explore) contest [7, 8] there are a kind of “laboratories” for experiments. There is also no mathematics itself, but constructive tasks and experimental results are directly connected with mathematics. This competition may be assessed as “experimental mathematics” or “experimental informatics”.

Mathematical Olympiads in Russia are generally held in one of two forms: oral or written.

Oral Olympiads generally inherit the rules and the structure of the oldest one, Saint-Petersburg City Mathematical Olympiad. There are some tasks given to the student. For each task she or he has three attempts to present a solution to jury member. If the solution is wrong, it should be explained to the student in the way that she or he understands her/his mistake but does not get any hint to the correct solution. If the student fastly corrects her/his mistakes, the presentation could continue without losing an attempt. In the final ranking only number of problems solved by the participant matters; neither number of failed attempts nor required time do not.

Of course, this form requires a lot of qualified jury members at the same time at the same place (about 5 participant per jury member) so it could be held for not so much students, at most for a few hundreds. This is the price that should be paid for a highest possible level of feedback. If there are thousands students that want to participate in oral Olympiad, the organizers have to use written preliminary stage. Another issue is that the very small amount of possible final results (from 0 to 7) which sometimes causes some troubles with assigning the prizes.

While at oral Olympiad the mark for the task is binary: the problem is either solved or not, at the written Olympiad more different marks are possible. At all stages of the All-Russian Olympiad each task is worth 7 points. Full 7 points is for a complete solution, 4–6 points is for a generally correct solution with some minor issues. 1–3 points mean that the problem is not solved, but the solution submitted has some ideas important for the certain task, for example, a correct answer is given without necessary explanation or some partial case is investigated.

This form is possible not only when all participants are gathered at the same place. For example, at the municipal stage of the national Olympiad (which differs from region to region) the participants are spread between dozens of schools in any region. At the distant stage of Euler mathematical Olympiad the students may solve the tasks at their homes and submit their scanned or photographed written works via the Internet.

There are a lot of different written Olympiads with different number and level of tasks. Some of them provide more points for more complex tasks and less points for the simple ones.

The third form, when only answer is submitted via the Internet, is used generally for the preliminary stages of Olympiads.

A written form seems to be the most flexible one. A participant can get for example 6 of 7 point for an almost-solved problem with a small arithmetical mistake. At the answer submission competition this result would be evaluated as completely unsolved task, the same issue we have at programming competitions when the formula with a one small mistake gives wrong answers on all tests. At the oral Olympiad a small mistake probably could be corrected during the presentation and the final mark for the task is the same as the mark for the student who initially solved it correctly.

From the other hand the student who just guessed a correct answer would not get so much points as in situation when only answer is need, but her/his solution would not generally get “zero” mark as at oral competition.

The Olympiad in Discrete Mathematics and Theoretical Informatics combines two forms of evaluation. There are presented the model of the mathematical object which are the most important for the informatics with constructive tasks offered to solve on these models. It is supposed that a good construction is a good result for the informatics; also a constructive task is considered to be a step for understanding and solving more theoretical ones and grasping the general properties of those objects.

2. ANALYSIS OF TYPICAL INFORMATIONAL TASKS

Mathematical and informatical tasks could be regarded from different points of view. Generally, teachers consider the ability to solve a problem as a learning goal. Exaggeration of this goal can lead to the effect of “training” when the student performs some actions senselessly or “unthoughtfully” and passes in a state of “mental stupor” if the task differs a little from familiar pattern. However, the problem has other more important dimensions. Any human activity can be represented as a sequence of solving certain tasks. When finding solutions to these problems, a person uses both existing precedents, templates for solving such problems, and constructs new, more efficient and universal ways of solving new problems based on those already known to him. While solving problems, the psychological features of a person are manifested, which can correlate with the peculiarities of his thinking style and with the chosen profession.

In some extent, the authors of the Olympiad tasks show interest in this view of the problem. In the sphere of our interests there are tasks and Olympiads in informatics and mathematics. About the difference between the mathematicians and informaticians Tseitin mentioned in his paper [4] (and comments to one of the authors of this paper): the mathematical style of thinking pushes the subject to search for a common concept through which to find the right solution tasks, if such an approach does not yield results, the subject begins to get nervous, loses interest to the task, and gives it up. On the other hand, “programmers” do not frighten the lack of a theoretical basis for solving problems and they easily move from one particular decision strategy to another.

It is interesting to analyze Olympiads in programming from the point of view of how the authors see the way of solving such problems.

Informatical tasks differ from mathematical ones by approach to solving: the theoretical model could be build and substantiated followed by a program written (without using special informatical knowledge); another approach is to describe the solution process using the restrictions in the task formulation and using a kind of optimized brute force.

Here is an example of the programming task with to possible ways of solving. This is the simplest task from the regional stage of 2018 All-Russian Olympiad.

In the school on informatics lessons students' answers are evaluated by integer marks from 2 to 5. The final mark is obtained as an arithmetic mean of all current marks rounded to the nearest integer. If the mean is exactly between two integers, it is rounded up.

All students aim to gain the final mark at least than 4 points. Unfortunately, a certain student gained a marks “2”, b marks “3” and c marks “4”. Now he plans to gain some “5” marks to reach the final mark at least 4 points. He needs to know which minimal number of “5” marks he must gain to achieve his goal.

You need to write a program which for any non-negative integers a , b and c gives the minimal number of “5” marks required.

There are two main approaches to solve this problem:

A first approach is to write a formula. We need to solve the inequality $\frac{2a+3b+4c+5x}{a+b+c+x} \geq 3.5$ and obtain $x \geq \frac{3a+b-c}{3}$. Then we need to find the minimal non-negative solution of this inequality and, if necessary round it up. The formula could be simplified as $x = (3a+b-c+2) \text{ div } 3$.

The second way is to make a loop calculating $\frac{2a+3b+4c+5x}{a+b+c+x}$ for x from 0 until x satisfy $\frac{2a+3b+4c+5x}{a+b+c+x} \geq 3.5$. However, authors of the problem artificially overcomplicate the problem allowing for a , b and c too great (and, in fact, impossible) values — up to 10^{15} . Brute force here becomes not efficient and the student has to use more efficient algorithm such as binary search. But this require to define preliminarily the maximal possible number of “5” mark needed. Authors suggest to use the upper bound $a+b+c$. Indeed, we have the arithmetic mean $\frac{(2+5)a+(3+5)b+(4+5)c}{2(a+b+c)} \geq \frac{a+b+c}{2(a+b+c)} = 3.5$.

Thus, on the first way students have to analyze the task completely. Arithmetical mistake can prevent them from getting even partial assessment of the solution. For example, the most common mistake in such problems is to write $x = (3a+b-c) \text{ div } 3$ as a minimal integer solution instead of $x = (3a+b-c+2) \text{ div } 3$. This gives us a correct answer for 1/3 of all cases, but the task will not be assessed positively, as the scoring is based on the passage of the solution (program) of a series of tests that are oriented not on the percentage of correct results, but on complete correctness for the values of parameters of different sizes, that is, the efficiency of the algorithms used:

For every subtask points are awarded only if all tests for this subtask and all required subtask are fulfilled correctly. Thus, despite that the formula is more efficient than either brute force or binary search, the solution completely fails due to small mistake.

At the same time we need mathematical reasoning even for “purely programming” solution, because for implementing the binary search algorithm we need first to obtain the upper bound for the parameter.

Thus, in simple problems, the authors allow the use of a “mathematical style of thinking”, that is, a theoretical approach to the problem and finding a theoretical solution (in this case, a formula) that gives an exhaustive solution to the problem. However, even in this case, the authors see another way of solution, which does not involve theoretical comprehension of the problem, but is based on the application of an algorithm that employs general type search strategies.

So, in programming contests the mathematical “beauty” of solutions is not taken into account, but rather algorithm efficiency. Also partial solutions are assessed.

Here is the reduced formulation of more complicated task:

There is the gas distribution system of one region, consisting of n nodes, some nodes are connected by one-way pipes. The node numbered 1 corresponds to the central gas storage.

A special robot is used to check the quality of the pipes. It is placed in a pipe system in one of the nodes and moves through the pipes (in the same direction in which gas is transmitted through the pipe), each time checking the pipe through which it moves. After making one or several movements between the nodes, the robot is extracted from the pipe system. Each launch of the robot satisfies one of the given specifications with different costs.

It is required to write a program that, by the description of the pipe system and the list of specifications, determines the minimum total cost of the robot launches, as a result of which all the pipes will be checked, as well as a list of required launches (on demand).

In this task building the full model is not possible during the Olympiad and only exhaustive strategies are taken into account, and the art of the participant is manifested in the extent to which an effective algorithm he will choose.

The authors of this paper see another way of combining these approaches in the Olympiad problems in mathematics and computer science. In this approach, instead of writing algorithms, the participant gets the opportunity to look for constructive solutions of an “algorithmic nature”, for example, to build a finite state machine or to describe a set with regular expressions.

In this activity, there are many features inherent in the “programming style”: there is the possibility of experiment and optimization of the solution, the solution can have a partial character and expand as the author’s understanding of the task increases.

On the other hand, in this approach, the “classical mathematical” problems are naturally included in the Olympiad, since the models on which participants “program” their solutions are a computer representation of some important concepts from discrete mathematics.

3. STRUCTURE OF THE DISCRETE MATHEMATICS AND THEORETICAL INFORMATICS OLYMPIAD

The Discrete Mathematics and Theoretical Informatics Olympiad attempts to combine the best features of traditional mathematical and informatical Olympiads, including different forms of evaluation. The problems are presented to participants through the Internet and their solutions are submitted the same way.

Tasks of the Olympiad are divided in two types: constructive and theoretical ones. Solving constructive task the participant has to build an informatical object, for example, a logical circuit, using a specific subject manipulator. The solution of a participant is evaluated according to a percentage of tests passed successfully.

Theoretical tasks require text solutions in the text-box. They are evaluated the same way as at written mathematical competitions: partial solutions get partial marks according to the criteria set by a jury.

The Olympiad is held in three stages. Both the first and the second stage are available from anywhere on the Internet. First stage is the training one. No prize is won and no point is awarded during this stage, it is used only to let the participants get acquainted with manipulators using some simple tasks as an examples. There are also no time limit and no restriction on attempts.

Second stage is the qualifying one. There are the same manipulators, but tasks are a bit harder. The duration of the second stage is usually two weeks, but participants have only three hours after logging on the system to submit their solutions.

Final stage is held in almost the same time in different cities. There are again the same manipulators but the tasks are again more harder than in the previous stage.

There are some problems the organizers of the Olympiad face during their work.

The first problem is that used computer models of theoretical informatics require some time to learn them. Participants have to avoid wasting time during the Olympiad to get familiar with the interface (note that in programming Olympiads analogue are programming languages that are learned in computer science lessons and knowledge of which is necessary to participate in the Olympics).

The problem is being solved in the following way: computer models of theoretical informatics, on which constructive solutions are built, are introduced at the training stage and do not change during the Olympiad season. Thus, the participant gets acquainted with the interface before the qualifying and final tour.

Second problem is that organization of the final round of the Olympiad requires the gathering of participants, but for Russia this is possible only for a small number of Olympiads and limits the number of participants due to financial problems.

The problem is being solved as the final stage is held in several dozen sites on the basis of universities that are as close as possible to the places participants' residences. Organizational problems on the sites are reduced as possible, since the Olympiad is held on its website, and site organizers should only check the participants' documents, provide access to the Internet and control the independence of their work.

The third problem is that there are usually 10 or more tasks in the final stage. Of course it could be hard for the students to distribute their time and attention between such a big number of different objects. However, in fact there are not 10 or more objects of investigation — tasks could be divided by three or four groups and in each group of task the participant faces the same object but need to explore it from different sides. For example, for one regular expression we can ask the following: build an expression itself; build a finite state machine; count the words satisfying this expression of the certain length; prove a statement about those words. Usually constructive tasks are followed by theoretical ones and could be regarded as a step for them: first a participant is allowed to "touch" a given object using a manipulator and then she or he is asked to make some proof or counting.

4. SUBJECT MANIPULATORS

At this time there are six subject manipulators within the framework of the Discrete Mathematics and Theoretical Informatics Olympiad: logical circuits, regular expressions, Tarski world, finite state machines, Turing machines and graph, and also additional module used for plain text tasks.

Logical circuits: a participant has to build a model of logical circuit which satisfy the conditions required. Logical gates could be drag-and-dropped into the box, their inputs and outputs could be connected by lines with each other and also with inputs and outputs of the whole circuit. While using the module a student can see how each of his gates is working on every input set. The circuit build by a participant is tested on all possible input binary sets.

Tarski world: a participant has to build a text expression corresponding to a predicata calculus formula using given variables and predicata, logical functions, quantifiers and brackets. The subject area is cellular square with a possible figures of different shape, color and size in some cells; those figures are possible values of variable. The predicata are constructed by the jury and may be different for different tasks, they involve figure properties listed above and also their relative positions. The expression build by a participant is tested on a number of "pictures" which are set by jury.

Regular expressions: this module allows to build regular expressions with three operations : concatenation ab , disjunction $a + b$ and iteration a^* . By the highlights a participant is informed which words satisfy the expression submitted and which ones do not; some of the words checked are pre-set by jury, others can be added by a participant oneself.

Finite state machines: this module allows to build either an accepting or a transducing finite state machine, depending on the task. After building a machine and clicking a "run" button, a

student can see how machine constructed by her or him works on a certain input word or on any another word the she or he submits.

Turing machines: a participant has to build a Turing machine by submitting a set commands. As in FSM module after building a machine and clicking a “run” button, a student can see how machine constructed by her or him works on a certain initial tape or on any another tape the she or he submits.

In all three modules above a solution, submitted by a participant is tested on a huge number of words which are set by jury by a regular expression.

Graph module: the module allows to build different graphs. There are options for adding/removing/markings vertices and edges, which could be turned on and off depending on needs of the certain task. A graph submitted is tested a script which is unique for any task.

Every manipulator module includes not only manipulator itself, but a textbox for comments as well. This textbox allows us to use the same module in two ways: for constructive and for theoretical tasks. Constructive tasks with are evaluated according to percentage of the well-passed tests while theoretical tasks solutions are read by jury members and evaluated by them manually.

5. EXAMPLES OF TASKS SUPPORTED BY MANIPULATORS

In this section we present three tasks from the Discrete Mathematics and Theoretical Informatics Olympiad, joined by common storyline, but with different questions and different manipulators. The is the one of general ideas of this Olympiad to let students look at the same object from different points of view and to investigate its different properties by different methods.

Problem 1. *There is a queue for voting. Members of the queue have figured out that for each of them at least one of his neighbours (before or behind him) votes “for”. Let “1” be for a person voting “for” and “0” be for person voting “against”. Build a regular expression describing all words corresponding to the queues that satisfy the condition above.*

To help the participant, there are some correct examples (1111, 0110 and 110011) and some counterexamples (1001, 01010, 10101 and 000) which are highlighted by red or green depending on whether they satisfy the current regular expression in the submission box or not.

Solution: of course there are more than one correct expression. One of them is $(011 + 11)(1^*(0011 + 011))^*1^*(1^* + 0)$. The word required starts either from 011 or from 11 depending on the vote of the first person in the queue. After that there could be any number of people voting “for”, i.e. any number of “1”. Then there could be one or two zeros, and then again at list two “1”. This pattern repeats until the end of the queue where could be again one person voting “against” or not.

Problem 2. *There is a queue for voting. Members of the queue have figured out that for each of them at least one of his neighbours (before or behind him) votes “for”. Let “1” be for a person voting “for” and “0” be for person voting “against”. Build a finite state machine accepting all words corresponding to the queues that satisfy the condition above.*

Solution: the solution could be found on the figure 1. The participant builds finite state machine using manipulator interface: he or she may add new vertices in the machine graph, arrows between them and choose letters on arrows.

S0 is an initial state. S1 is a state when current symbol is “0” and the following symbol is definitely “1” (i.e. when current symbol is the first symbol in the word or the second “0” in “00”).

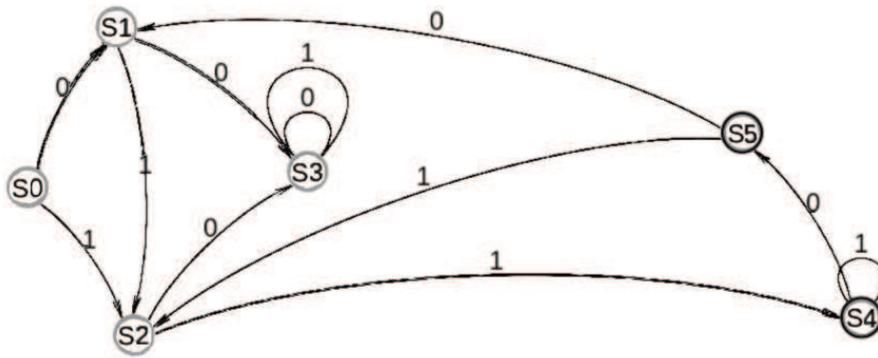


Figure 1. Finite state machine — solution for the problem 2.

S2 is a state when current symbol is “1” and the following symbol is definitely “1” (i.e. when the previous symbol is not “1”, but “0” or nothing).

S3 is a state reached by the machine when getting a symbol which does not satisfy the required conditions, i.e. when machine gets a word like 00..., 10..., ...000... or ...010.... The machine cannot leave this state.

S4 and S5 are the final states corresponding to the words with “1” and “0” at the end. If the machine finishes its work in this state, it means the current word is accepted. If it is not the whole input word, the machine continues its work and can leave these states.

Both this and the previous tasks solutions are tested on all words in alphabet {0, 1} with length from 1 to 10. The final mark is proportional to number of words for which a participant solution gives the same result as the reference one.

Problem 3. *There are n people in the queue for voting. They have figured out that for each of them at least one of his neighbours (before or behind him) votes “for”. Prove that the amount of people voting “for” is at least half of the total amount of people in the queue.*

Solution: for each people voting “against” we match him to a person voting “for” by the following rule: if he has the only neighbour voting for, we match him to this only neighbour; if he has two such neighbours, we match him to any of them. Note that we could not match two different people voting “against” to the same person voting “for”, otherwise this person voting “for” has two neighbours voting “against” which is impossible. So for any person voting “against” we have unique person voting “for”, which means the number of people voting “for” is equal or greater than number people voting “against” which immediately implies the statement required.

Unlike two previous tasks, this one requires text solution and is evaluated manually.

6. SOLUTION ANALYSIS

Although problem 1 appeared to be rather simple of three ones regarded in this paper according to the total number of points awarded, there were only two complete solutions.

Another 12 solutions got the mark of 95 % or more, which means they work correctly at least on 95 % tests. They failed on different tests, but the most common mistake for almost correct solutions was evaluating the word 01110 as wrong, probably just because it has no zeros in the middle.

The marks of this task solutions showed us a continuous spectre of values from 75 % to 100 % with totally 55 solutions in this range. Also there were 4 solutions from 50 % to 75 % and 7 ones from 0 % to 15 %.

Mathematically tasks 1 and 2 are equivalent as there is an isomorphism between building regular expression and building accepting final state machine problems.

Problem 2 appeared to be a bit more simple than problem 1, but not so much. Totally we have 8 completely correct solutions. Another 6 solutions got 95 % mark or greater which gives us exactly the same number of people in 95–100 % range as for previous problem.

The marks of this task solutions showed us a continuous spectre of values from 70 % to 100 % with totally 60 solutions in this range. Also there were 4 solutions from 50 % to 70% and 6 ones from 0 % to 10 %.

Only 8 of 73 submitted solutions of the problem 3 were evaluated as completely correct. These solutions were based on some different ideas.

One solution was based on the idea that any person voting “for” provides himself as a neighbour at most to 2 other persons, so if we need n neighbours voting “for”, we must have at least $\frac{n}{2}$ people voting “for”. Another participant regarded separately people on odd and even positions in the queue and noted that in any of these two sequences we cannot have two people in a row voting “against”.

Third possible idea was to divide the queue on subqueues of 4 person and a reminder. This idea was used in some solutions, but it requires more precise investigation of the case of 1-person reminder which was missed in some solutions of this type.

A huge amount of solution was based on regarding “the worst case” of the sequences like 011001100..., but most of them were missing the explanation what does “the worst case” actually mean and why it is really the worst. Such solutions were partially evaluated.

There were totally 11 solutions were evaluated as almost correct and got 2 or 2.5 of maximum 3 points. As partial solutions described above, they missed some cases or part of explanation.

Also 18 solutions got marks 0.5 or 1 as they included important ideas but were very far from complete solution.

Thus the theoretical problem appeared to be more complex than a constructive ones despite that its formulation is even more simple and its understanding do not require. For the whole Olympiad showed the correlation between marks for the constructive and theoretical tasks about 42 %.

7. CONCLUSIONS

The use of computer models of the concepts of discrete mathematics and theoretical informatics provides the basis for creating a wide range of constructive problems. The solutions to these problems allow automatic verification, which can also be used as a feedback reaction for participants, helping them conduct experiments that suggest a constructive task. Verification of partial solutions can be carried out on test suites (as in the programming Olympiads), but in some models it is possible to fully test solutions if it is theoretically possible (above there should be examples of a Turing machine and finite state machines, with comments that in both models a part checking is done on the generated data, but in the second one you can perform a full check).

The use of computer models of the concepts of discrete mathematics and theoretical computer science creates the conditions for creating banks of problems that have different content,

but are based on a specific mathematical model. Such task banks can be used by teachers, in the organization of group work, and by university teachers who teach courses in discrete mathematics and theoretical computer science.

In the work it is shown that constructive problems are more accessible for participants of the Olympiad than theoretical ones, but there is a significant correlation between numbers of the solutions of those and others. Therefore, the enrichment of the Olympiads by constructive tasks, on the one hand, expands the number of their possible participants, on the other, creates conditions for a better understanding of important substantive ideas that theoretical tasks carry on.

References

1. Order of ministry of education and science of Russian Federation at 30.08.2017 No866 "Of setting the list of Olympiads of schoolchildren for 2017/18 school year. URL: <https://rg.ru/2017/09/27/minobrnauki866-dok.html> (in Russian).
2. Okulov S. M.: Informatics. Schoolchildren intellect development. Moscow: BINOM. Laboratory of knowledge (2005) (in Russian).
3. Anisimova S. A.: Discoveries are made due to the talent of singles! Scientific and methodical electronic magazine "Concept" (2013) (in Russian).
4. Tseitin G. S.: HEXPLORE. Computer Tools in Education vol. 6, pp. 84–87 (1998) (in Russian).
5. van der Wegt W.: Bridging the Gap Between Bebras and Olympiad: Experiences from the Netherlands. Olympiads in Informatics, Vol. 10, pp. 223–230 (2016).
6. Dagine V., Stupuriene G.: Bebras – a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking Informatics in Education, vol. 15, no. 1, pp. 25–44 (2016).
7. Pozdniakov S. N., Posov I. A. Puhkov A. V., Tsvetova I. V.: Science Popularization by Organizing Training Activities Within the Electronic Game Laboratories. International Journal of Digital Literacy and Digital Competence, 3(1). January-March. pp. 17–31 (2012).
8. Maytarattanakon A., Posov I. A.: Automation of distance contests based on research problems in mathematics and informatics. Computer Tools in Education, vol. 6, 45–51 (2014) (in Russian).
9. Wing, J. M. Computational thinking. Commun. ACM 49, 33–35. (2006).
10. Wing, J. M. Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 366 (1881): 3717. (2008).
11. Papert S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, Inc., Publishers / New York (1980).
12. Pozdniakov, S., Gaisina, S.: New trend in Russian informatics curricula: integration of math and informatics. Local proceedings of the 7th International Conference on Informatics in Schools: Situation, Evolution and Perspectives, p. 91-100 (2014).
13. Akimushkin V. A., Pozdniakov S. N., Chukhnov A. S.: Constructive Problems in the Structure of the Olympiad in Discrete Mathematics and Theoretical Informatics Olympiads in Informatics, Vol. 11, pp. 3–18 (2017).
14. Baker E., Dickieson J., Wulfeck W., O'Neil H.: Assessment of Problem Solving Using Simulations 1st Edition Rutledge (2016).
15. Fan-Ray Kuo, Nian-Shing Chen, Gwo-Jen Hwang: A creative thinking approach to enhancing the web-based problem solving performance of university students Computers and Education, vol. 72, pp. 220-230 (2014).

Received 20.02.2018, the final version — 04.04.2018.

Компьютерные инструменты в образовании, 2018

№ 2: 55–67

УДК: 373.62 : 004.5

<http://ipo.spb.ru/journal>

АНАЛИЗ ПОНИМАНИЯ МАТЕРИАЛА ТЕОРЕТИЧЕСКОЙ ИНФОРМАТИКИ В КОНКУРСАХ И ОЛИМПИАДАХ ПО ИНФОРМАТИКЕ

Поздняков С. Н.¹, Чухнов А. С.¹, Паньгина Н. Н.²

¹ Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»
им. В.И. Ульянова (Ленина), Санкт-Петербург, Россия

² Муниципальное автономное образовательное учреждение дополнительного образования
«Центр информационных технологий», Сосновый Бор, Россия

Аннотация

Конкурсы и конференции следует рассматривать не только как способ сравнения уровней подготовки участников по предмету конкурса или олимпиады. Их можно рассматривать как иной способ организации умственной деятельности участников. Различные типы задач инициируют различные виды интеллектуальной деятельности у учеников, олимпиадные и конкурсные задачи отличаются от «школьных» нестандартностью, сочетанием доступности формулировки и неочевидностью решения, возможностью различных подходов к решению задачи.

В то же время, разнообразие конкурсов и олимпиад ставит задачу их изучения и классификации с точки зрения поддержки различных стилей мышления. В данной работе будет сделан анализ конкурсов и олимпиад по информатике, направленный на выявление аспектов интеллектуальной деятельности ученика, оценка которых традиционными «школьными» средствами проверки недостаточно информативна, в то время как организация и отслеживание внеурочной деятельности позволяет лучше оценить развитие интеллектуальных механизмов ученика без применения тестовых технологий.

Олимпиада по Дискретной Математике и Теоретической Информатике находится на стыке двух наук. В статье рассматриваются проблемы, с которыми сталкиваются организаторы соревнований по математике и информатике и демонстрируются некоторые способы их решения на примере указанной олимпиады.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 18-013-01130).

Ключевые слова: математические олимпиады, олимпиады по программированию, конструктивные задачи, дистанционное обучение, конструктивная деятельность, компьютерные средства, предметные манипуляторы.

Цитирование: Поздняков С. Н., Чухнов А. С., Паньгина Н. Н. Analysis of the Understanding of the Material of Theoretical Informatics in Competitions and Olympiads in Informatics // Компьютерные инструменты в образовании. 2018. № 2. С. 55–67.

Поступила в редакцию 20.02.2018, окончательный вариант — 04.04.2018.

Поздняков Сергей Николаевич, доктор педагогических наук, профессор кафедры ВМ-2 СПбГЭТУ «ЛЭТИ»; 197376, Санкт-Петербург, ул. Профессора Попова, д. 5, корп. 3, кафедра ВМ-2, pozdnkov@gmail.com

Чухнов Антон Сергеевич, старший преподаватель кафедры ВМ-2 СПбГЭТУ «ЛЭТИ», septembreange@gmail.com

Паньгина Нина Николаевна, Заслуженный учитель РФ, методист МАОУ ДО ЦИТ, учитель информатики МБОУ «Лицей 8», nina_pangina@mail.ru

Sergei N. Pozdniakov,
professor of Department of Mathematics,
Saint-Petersburg Electrotechnical University;
197376 Saint-Petersburg, ul. Professora
Popova 5, korp. 3, Dep. of Mathematics,
pozdnkov@gmail.com

Anton S. Chukhnov,
lecturer of Department of Mathematics,
Saint-Petersburg Electrotechnical University,
septembreange@gmail.com

Nina N. Pangina,
Honored teacher of Russian Federation,
Methodist of Municipal autonomous
educational institution of additional
education "Center for Information
Technologies",
nina_pangina@mail.ru



Our authors, 2018.
Наши авторы, 2018.