Olympiads in Informatics, 2018, Vol. 12, 147–157

DOI: 10.15388/ioi.2018.12

PRASK – an Algorithmic Competition for Middle Schoolers in Slovakia

Michal ANDERLE
Faculty of Mathematics, Physics and Informatics of the Comenius University Bratislava, Slovakia,
e-mail: anderle.michal@gmail.com

Abstract. Although informatics education is compulsory in Slovak middle schools, its curriculum is insufficient to prepare the pupils for competitions such as the national Olympiad in informatics. There is a huge gap in knowledge that needs to be addressed. To reduce this problem, PRASK, an algorithmic contest for middle schoolers (approx. ages 10–15), was created in 2015. In this paper we discuss the main concepts behind this competition, ways of dealing with insufficient computational knowledge, types of tasks used to engage and educate young competitors, and some results and lessons learned from the first four years of this competition.

Keywords: algorithmic competition, middle schools, PRASK.

Introduction

Computer science is a rapidly evolving discipline, therefore schools are struggling to reflect this progress in their education process. In addition to this fact, computer science is mostly targeted at high school students, and curricula for lower grades has been emerging only recently.

From the very beginning, education of computer science for gifted pupils was taken over by universities, various competitions and volunteer organizations. This trend has been present in Slovakia for more than thirty years. However, the largest algorithmic competitions in Slovakia focuses only on high school students.

This brings up the question whether a similar format of competitions could be used in middle schools. In this article we will introduce such competition, PRASK, point out changes that were necessary to incorporate in order to adapt the competition to a new audience and also problems that occurred.

Algorithmic Competition PRASK

History of Correspondence Seminars

Dagienė and Futschek (2010) believe that many countries are lacking high-quality computer science teachers who would be able to introduce the pupils to the computer science in an interesting way. It is therefore becoming more and more common that instead of developing algorithmic skills, the use of specific software applications is trained.

However, many authors believe (Dagienė and Futschek, 2010; Dagienė *et al.*, 2015; Dagienė and Stupurienė, 2016; Forišek and Winczer, 2006; Kalas and Tomcsanyiova, 2009; Kubica and Radoszewski, 2010) that by means of informatics competitions, we can present various parts and used concepts of computer science. Competitions like Olympiad in informatics, Bebras and many others have been the result of this line of thinking. And also many years of experience show that such a way of presenting computer science can be very effective.

Slovak Republic has a rich history of algorithmic competitions. The Olympiad in informatics was established in 1985 and has been educating talented pupils for more than 33 years. There is even older competition, Correspondence seminar in programming (KSP), created in 1983. This competition serves as a stepping stone for all pupils interested in computer science, primarily high school ones.

Correspondence seminars organized for high school students have a long tradition in Slovakia and the Czech Republic, not only in computer science but also in mathematics and physics. The original purpose of the seminars was to educate talented pupils in the natural sciences. Another reason for its creation was also the above-mentioned shortage of qualified teachers. Unfortunately, this has not changed in 35 years (Forišek, 2007), and this form of non-formal education remains key in teaching of gifted pupils. Even seminars for middle schools have been appearing.

Origin of the PRASK Competition

There were multiple reasons for creating algorithmic competition for middle schoolers and they all came together in 2015. Perhaps the most significant one was the feeling of the original founders that pupils are getting to algorithmization and programming relatively late, if at all. This negatively affected competitions like KSP and Olympiad, which lacked young competitors.

It is important to note, the pupils' interest in algorithmization and programming did exist, and it has also seemed that basic algorithmic concepts could be taught even before high school. KSP organized multiple programming schools where these presumptions were confirmed. Moreover, for mathematics and physics, there were already successful equivalents of high school seminars for middle schoolers, so the PRASK competition was created.

First of all, objectives of this competition and the basic assumptions had to be clarified. PRASK was intended for talented pupils of middle schools interested in computer science. However, the middle school in Slovakia consists of five years (10–15 years old) in which the knowledge of pupils changes considerably. Consequently, the competition primarily focuses on the last three years of middle school. But of course, involvement of the younger pupils is always welcomed. This restriction determines the degree of mathematical knowledge that is to be expected from all competitors, based on the Slovak national curriculum.

However, organizers do not assume any common knowledge in field of computer science. From personal experience and interviews with pupils, it was clear that different schools taught different things and used different environments and tools. Therefore, the organizers had to assume that the prior knowledge of pupils in computer science will largely vary.

The main goal of the PRASK competition was promotion and development of algorithmic and programming skills. The contest was meant to reach out to a complete beginners who had not came across programming and algorithms whatsoever, but at the same time provide considerable challenge to more experienced competitors. That should provide scaffolding to the high school competitions like KSP and Olympiad in informatics.

Format of the Competition

The format of the PRASK competition was strongly inspired by KSP. Annually, two parts of the competition are held, each consisting of two rounds. Within one round, five tasks are published. There are no performance categories, the tasks are assigned for everyone. Each round has a set deadline for the pupils to submit their solutions. There is at least one month between the release of the tasks and the final deadline.

Each task has it own means of solution – some require uploading a program, other written description of the solution. After the deadline, the pupils' solutions are graded and the feedback is sent back in the form of a comment. Feedback contains commendation for good work, explanation of mistakes and grading, and some follow-up questions to the task.

At the end of both parts, the week long camp is organized for top 18 competitors. These camps consist of algorithmic lectures, but also sports, games and team building activities. Program of the camp is focused on presenting computer science as a fun and interesting topic, and on building a community of young people interested in it. This community building aspect is very important, because it ensures continuity (Forišek and Winczer, 2006).

Tasks

The PRASK contains three types of tasks – theoretical, practical and programming. Each type covers different area of informatics and ultimately helps developing different abilities. Each round contains two theoretical, two programming and one practical task.

Programming Tasks

One of the main goals of the PRASK is to develop pupils' programming skills. Therefore, each round contains two programming tasks. In these tasks, competitors need to write and debug a program that solves assigned task. The uploaded solutions are automatically tested and the competitor finds out his/her results immediately. The solutions can be then reworked and submitted again until the deadline.

The tasks often contain multiple easier subtasks worth partial points and the whole process is similar to the one at the IOI. Solutions can be written in several programming languages, Python and C++ are most frequently used.

The difficulty of programming tasks is determined by a model solution. It should only use basic concepts – variables, cycles, conditions and arrays. In harder tasks, it is also possible to use algorithms and data structures that are implemented in the used languages, most commonly sorting and binary search trees. Model solutions should not require knowledge of any advanced algorithms or data structures, which usually excludes graphs. The recursion is also avoided.

However, considering these limitations, solutions require some non-trivial idea that leads to a more effective solution. These tasks should not be just straightforward implementations. But as mentioned above, subtasks are commonly used, meaning that trivial or slower solutions always score at least some points.

It is clear that this type of task would be inappropriate for beginners. PRASK should be accessible to all pupils interested in computer science regardless of whether they know how to use a programming language or not. Hence, the programming Hatchery was created. It consists of four sets of tasks and study texts that create a tutorial to C++. In these four sets, pupils learn to use variables, conditions, cycles and arrays. And because this competition would require much more effort from beginners (first learn programming language in Hatchery and then solve problems), points obtained in the Hatchery are able to replace points from programming tasks. Each set of tasks in Hatchery can replace one task in PRASK, which means that beginner contestant can learn programming in half a year and then use this new knowledge to later solve tasks.

Finally, we add that the choice of C++ is purely practical. Besides the fact, that organizers themselves have more experience with this programming language, and they were able to use existing materials when creating the Hatchery, the C++ have some advantages over Python, e.g. support at IOI and Slovak national competitions, due to its speed.

Practical Tasks

Practical tasks are often interactive and present new technology or part of computer science to the pupils. Their main goal is to promote IT and to motivate pupils to further work. The interactivity and the unusualness of these tasks is an attraction for contestants and indeed, this type of task is the most popular one. Presented problem is often a puzzle, that must be solved and the feedback is immediate.

A nice example of such task is the very first task, in which pupils were referred to a purely black web page containing a secret password. Pupils had to figure out that page contains images of black letters, so the background of the page needs to change. It was up to them to use Javascript, view the source code of the page, edit it locally or use entirely different approach.

During the four years of this competition, pupils had to use Google, Excel, Word or various image editors in new, inventive ways. Practical tasks also include problems, that need to be solved by using specific technology, for example AutoHotKey for automatic mouse and keyboard control, SQL or Prolog. For these problems, a quick tutorial is presented, containing the necessary concepts and commands. Contestants need to learn how to work with them and use them effectively. To help them do that, interactive web environments and easy subproblems are offered.

Even though the tasks might be at times technologically challenging, it seems that even such tasks do not discourage contestants. In the survey, that was performed in January 2018, nearly 93% of participators stated that they had little to no technical problems while competing in PRASK. This significant percentage is probably achieved by the existence of tutorials that are applicable to different systems (Windows, Linux, Mac), as well as the involvement of the organizers who can be contacted by the pupils at any time.

Theoretical Tasks

Probably the most unconventional and challenging to prepare are theoretical tasks. These tasks direct contestants to design some of the known algorithms or dive deep into specific area of computer science. There were tasks based on well-known algorithms or data structures (spanning trees, binary search trees) but formal languages and automatons are commonly used as well. Previous knowledge is not necessary, organizers even assume that presented problems are not known by competitors, task statement contains all important rules and relationships, contestant needs to combine them, come up with their new usage and formalize their thinking.

For this type of problem, participants submit a text document describing their solution. Description needs to contain not just an answer, but also approach they used to get the answer. Some sort of formalized “algorithm” is often required as well, either in pseudo-code or natural language.

Theoretical tasks are probably most different from practises used at IOI but are actually very reminiscent to the format of Slovak national olympiad and KSP. It focuses on thinking process, probably the most important ability to develop. As Michal Forišek stated (Forišek, 2007) "practice only" competition style (such as IOI) are restricted by actual implementation and marginal issues such as debugging techniques, library knowledge, etc. Slovakia is a rather small country and the number of participants in this type of competition peaks at one hundred. Hence, very personal approach can be used.

In PRASK, this type of task lacks immediate feedback, solutions are corrected after the deadline. But benefits include more personalized and detailed feedback and focus on the thinking process of the contestant. It is not even uncommon to have different expectations on different contestants and score them adequately to it. Even though PRASK presents itself as a competition, the personal growth and education of participants is even more important.

Unfortunately, theoretical tasks are the least popular among the contestants. The main reason for this is the need to write down solutions. Up to 50% of contestants said that they were discouraged by it. This result is not surprising, write down solution takes most of the time and pupils are not used to it. On the other hand, we believe that this part of competition is beneficial for the pupils, improving their ability to express and write understandable procedures and algorithms.

Concepts used for theoretical task sometimes include one stroke drawing, various dynamic programmings, regular expressions, error detection and correction codes, logic gates, deterministic finite automata, mergesort, minimal spanning tree, Euclid's algorithm and others.

Experience and Challenges for the Future

Preparation of Theoretical Tasks

Because theoretical tasks are the most atypical and also the most interesting and challenging to prepare, we will discuss their preparation in more detail. As we mentioned before, theoretical tasks are used in the Slovak national olympiad as well as in correspondence seminar. However, they use is a bit differently. Most of the times they are classic tasks, but instead of implementation, contestants need to write down a description of solution. They rely on existing knowledge, often using classic algorithms and data structures. These tasks cannot be used in PRASK, where the tasks should be solvable from the scratch. This forces organizers to create entirely new, original tasks fulfilling all the assumptions that are put into them. In fact, we are not aware of any other competition that would be using similar tasks.

All of this put a lot of responsibility on tasks seters. In order for these tasks to be solvable and understandable, a suitable form must be used. Tasks cannot use technical terms, problem description uses a story that presents a problem in a more comprehensible form. Similar stories are used at IOI, but their part in PRASK is more crucial. This story offers a metaphor with which pupils can work more easily. If they never encountered graph problem, we cannot use terms as vertex or edge, we need to present e.g. cities interconnected by roads. Note, that choice of a suitable metaphor is really important, helping pupils to work more efficiently with original abstract concepts and even leading them to the right solution (Forišek and Steinová, 2013). Additionally to metaphor, theoretical tasks tend to include images or interactive environments pointing to various special cases that may occur. That also helps pupils to better understand presented problem.

To create problems that are accessible for beginners and challenging for experienced contestants, theoretical tasks are divided into several, successively more complicated, subtasks. These subtasks are designed to progressively guide participants towards the solution. The first subtask presents specific inputs and contestants can solve them using just pen and paper. While solving these easy subtasks, pupil will develop some sort of strategy or algorithm. Next subtasks ask the pupil to formalize this algorithm and formulate it in natural language. The hardest subtasks, meant for the experienced contestants, often ask for a proof of correctness.

The last issue the PRASK contestants have to deal with is an estimate of time complexity. It is a basic principle that must be taken into consideration when designing an efficient algorithm and the means by which the solutions will be evaluated. Therefore, efficiency must appear in the statement in some form. Most of the time, intuitive view of complexity is used, asking from pupils to create algorithm that would be feasible even with pen and paper on larger inputs. From this description is obvious that backtracking all possibilities is correct but not fast enough. Alternatively, in some problems a number of specific steps can be used. For example, when sorting, pupils were told to minimize the number of comparisons.

Involvement of High School Contestant

The Correspondence seminar in programming is organized by university students at Comenius University and is targeted at high school students. Since the PRASK is intended for middle schoolers, high school students can be involved in its organization. In fact, half of the current organizers do not attend university yet.

The selected contestants of KSP are offered the opportunity to participate in the preparation of PRASK. This selection is based on knowledge of computer science, age and personal interest. These young organizers are preparing a vast part of competition, writing statements and solutions, preparing test data, and planning camp activities and lectures. University organizers serve as their mentors, check quality of the prepared materials and give feedback. Also tasks ideas originate mostly from senior organizers.

This cooperation is very interesting and fulfilling for both sides. Seniors have the opportunity to share their experience and high school students can improve in several different skills. On the one hand, their understanding of computer science arises. Being able to prepare and present lecture or create test data for algorithmic problem needs deeper understanding of underlying concepts. On the other hand, in these activities are also involved important soft skills. Write a clear task statement, come up with a fun idea for camp game and implement it, and take responsibility for the participants are qualities, that all will useful in their future lives.

It also helps with continuity. Many of these students will continue to be involved in similar activities during their university years. Either in PRASK, KSP or the Slovak national Olympiad. Helping to prepare them from young age is beneficial to all of these competitions.

Statistics

In this part, we will take a look at some data from the first three years of PRASK competition. First, let's look at the number of contestants in each round, shown in Fig. 1. It is important to note, that these numbers can be misleading. Rounds 1 and 2, as well as rounds 3 and 4, in each year belong to one part and they are scored together. Therefore, number of contestants in round 2 (or 4) is union of contestants in round 1 and 2 (3 and 4).

But, there is a part of contestants that only solved few easy subtasks of some interactive problem, but overall they got very few points. Therefore, in Fig. 2 we will show the number of contestants with at least half the points. We believe that it is possible to get at least half of the points with a bit of effort and these participants are therefore reasonably engaged in the competition.

Notice, that the number of contestants peaked during round 1 in 2015 and round 3 in 2016. In these two rounds, instead of programming tasks, the interactive and playful

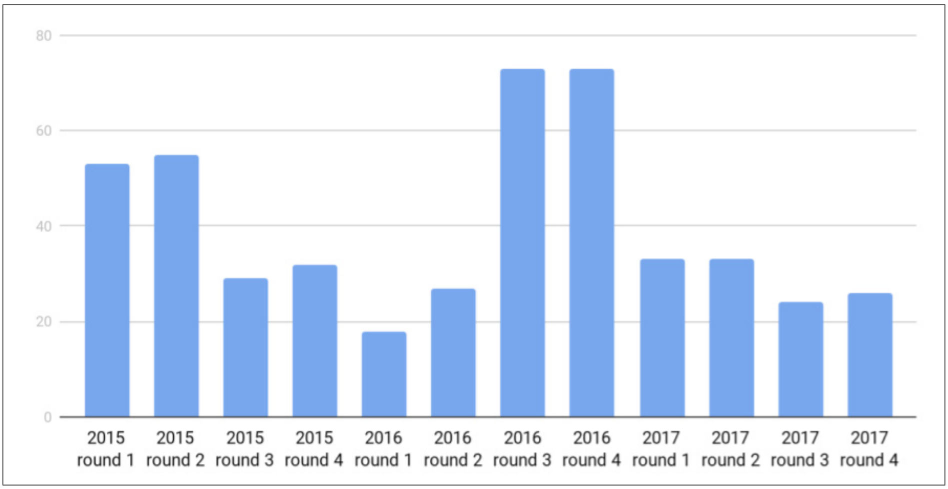

Fig. 1. Number of contestants in PRASK competition.

self-designed environments were used. Both of them were very popular, which resulted in a higher participation rate.

The average number of points for contestant is 35.58 out of 75 possible. However, if we take a look only at participants with at least half of the points, it grows to 56.44 which is more than 75%. In the Fig. 3, we can take a look at the average percentage for each tasks published during first three years. Notice, that almost all of them range between 50 and 75 percent. We can assume that the tasks are reasonably difficult and it is fairly easy to get substantial part of the points, mainly thanks to the easiest subtasks.

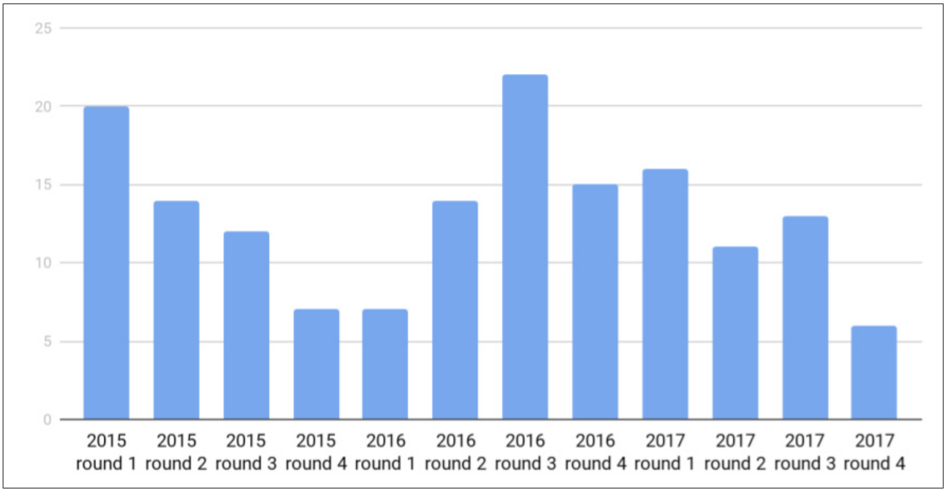

Fig. 2. Number of contestants with at least half the points.

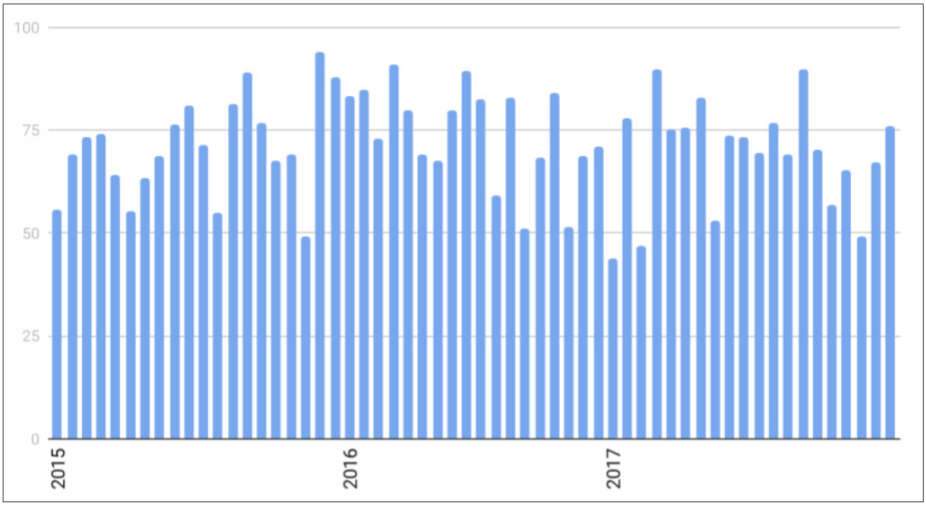

Fig. 3. Average percentage in individual tasks.

Finally, we will mention two interesting things that are present in scoreboards. The first is that girls took a rather dominating spot. Although the majority of contestants are still boys, first three positions during the three years were occupied only by girls. The second interesting point is that PRASK has had some very young competitors. There have always competed at least one fifth grader (age 10–11) and the youngest contestant was a girl from fourth grade (10 years old, not even in middle school).

Future Challenges

PRASK is now being organized for almost four years and the fifth year is being prepared. It is still evolving and there are several issues that need to be addressed.

Probably the most important is low participation count. On average, there are around thirty participants in each round, a number we would like to increase. Since PRASK is still fairly young, it is not well known among pupils or teachers. Teachers are the target category we would like to focus on at first, as they have great impact on huge number of students.

Other problem PRASK is facing is creation of consistent problem sets. Organizers are not sure about intended complexity of the tasks, which leads to trial and error approach. Also, it is really hard to prepare high quality theoretical tasks, which poses the biggest challenge in problem setting. Theoretical tasks also need to be popularized among contestants. Using interactive environments would be the best, as the survey showed, but those environments are time consuming to prepare, so the fine line needs to be found.

Conclusion

We presented PRASK, algorithmic competition for middle schoolers, and looked at the experience gained from the first four years of organizing it. This competition does not imply any prerequisites on previous algorithmic knowledge, trying to offer pupils an environment in which they can grow. The tasks are trying to concentrate on the idea of the solution rather than the actual implementation.

It turns out that pupils have fun dealing with interactive tasks in which they receive immediate feedback or tasks in the form of puzzles. In our opinion, theoretical tasks have tremendous potential but they are not as popular because of the need to write down description of the solution. For the future, the greatest challenge is to increase participation rate and improve pupils' interest in theoretical tasks.

References

Dagienė, V., Futschek, G. (2010). Introducing informatics concepts through a contest. In: *IFIP Working Conference: New Developments in ICT and Education. Universite de Picardie Jules Verne, Amiens, 2010*.
Dagienė, V., Pelikis, E., Stupurienė, G. (2015). Introducing computational thinking through a contest on informatics: Problem-solving and gender issues. *Informacijos Mokslai/Information Sciences*, 73.
Dagienė, V., Stupurienė, G. (2016). Informatics concepts and computational thinking in K-12 education: A Lithuanian perspective. *Journal of Information Processing*, 24(4):732–739.
Forišek M. (2007). Slovak IOI 2007 team selection and preparation. *Olympiads in Informatics*, 1, 57–65.
Forišek. M., Steinová, M. (2013). *Explaining Algorithms Using Metaphors*. Springer.
Forišek, M., Winczer, M. (2006). Non-formal activities as scaffolding to informatics achievement. Information. *Technologies at School*, 529–534, 2006.
Kalas, I., Tomcsanyiova, M. (2009). Students' attitude to programming in modern informatics. In: *Proc. 9th WCCE 2009, Education and Technology for a Better World*.
Kubica, M., Radoszewski, J. (2010). Algorithms without programming. *Olympiads in Informatics*, 4.

M. Anderle is a PhD student at the Comenius University in Slovakia. Since 2011, when he joined the university, he has been involved in the organization of various algorithmic competitions, mostly the Slovak national olympiad and Slovak correspondence seminar in programming, and summer schools. He is also a cofounder of the algorithmic competition for middle schoolers – PRASK.