Croatian Journal of Education
Vol: 13 (3/2011), pages: 201-231
Professional paper
Paper submitted: 28th March 2011
Paper accepted: 10th December 2011

THE ROLE OF COMPETITIONS IN COMPUTER SCIENCE CURRICULUM

Ljerka Jukić and Ivan Matić
Department of Mathematics,
Josip Juraj Strossmayer University of Osijek, Croatia

ABSTRACT

In the recent years, competitions in school subjects have slowly but surely occupied a prominent place in the educational process and now play an important role in motivating and inspiring students to enhance their learning. This paper presents a brief overview of the competitions in computer science in Croatia, highlighting their strengths and weaknesses and stressing their importance and application in the methodology of teaching.

Keywords: *competition in computer science, development of social skills, teaching methods in computer science.*

INTRODUCTION

Competitions have long served as an important educational method in pedagogy and methodology of education. According to (Vukasović, 1994:370) competition is a “form of activity in which individuals or groups of people compete to accomplish the task faster and better than the other individuals or groups which are entrusted with the same task, and thus it [the competition] increases commitment and devotion to work. Competition as an educational tool has a very powerful meaning. It influences the competitors’ feelings, introduces certain liveliness and cheerfulness among the competitors and develops an interest in the new knowledge, raises the competitive spirit and the desire to win. Thus, it encourages success, action, by demanding that given tasks be performed on time, as soon as possible and as well as possible. "

Following a rapid development of technology and infiltration of computers into everyday life, competitions in computer science in the Croatian educational system have been developing at the same quick pace in the past 15 years. Competitions in this school subject are equally developed and popular in other countries; in the highly developed as well as the slightly less developed ones.

It will be seen how the diversity of Croatian students and their results do not fall behind their peers in other countries.

In Croatia, the first competitions in computer science took place in the early 90s in the last century, when numerous problems had preceded the organization of each of these events. These problems were primarily of technical nature, because in many parts of the country it was a real challenge to form a place with a sufficient number of computers on which students could work. In addition, owning a personal computer was relatively rare at the time and it was necessary to ensure the equipment needed for the competition in advance. The need for special equipment is present in other subjects, such as chemistry or physics, but schools have been equipped with the teaching materials for these subjects for the past 40 years, which is a significantly longer period of time than has been the case with computers.

In 1992 and 1993 computer science started to be introduced into the school curriculum as a full-fledged subject. Few schools possessed adequate computer equipment at that time. In those early days, most computer science competitions took place in the specialized centres and technical education clubs, which supplied the technology and the sufficient number of the appropriate computers. From the very beginning, it was clear that students should work on the matching equipment and personal computers, and should not use different computer brands such as Spectrum, Atari, Commodore, Amiga, etc. Thus, until the second half of the 90s, the competitions in computer science in Osijek and Osijek-Baranya county were held simultaneously in several places.

The development of computer equipment, a significant number of donations of IT technology in the 90s, and the increased interest in the computer science competitions enabled a successful resolve of the initial problems. Owning a personal computer ceased to be a luxury and became a necessity, and today many students posses their own personal computers. The number of competitors has increased greatly over time and competitions are still being held at multiple locations simultaneously, thereby reducing the possibility of data leakage.

Computer science competitions differ from the competitions in other subjects in offering students the possibility to choose the approach which suits them best:

- problem solving - students (mostly individually) solve pre-made tasks, where they are required to develop programs (applications) that will work as quickly and as accurately as possible
- software fairs - the students work individually (teamwork is also allowed) to make an application which is presented to the expert board. Students are not given any rules, they can choose a theme and a programme language they wish to work in when creating the application. Unlike the first approach, this one allows the ultimate development of students' creativity. The time scheduled for making the application is not limited.

The presence of the evaluation board that assesses the presented application contributes to the subjectivity of the evaluation. In order to reduce the subjectivity, the assessment criteria are predetermined and competitors are familiarized with them as well. The assessment criteria for the presented software are categorized in the following manner:

- the originality of the idea (e.g. the improvement of an existing programmes or a user application of educational nature),
- creativity, performance and adaptation of the application to the user ,
- possible errors or imperfections in the functioning of the software (when working with a large number of data or on a weaker PC)
- software solution (the use of algorithmic knowledge, simple or complex solutions, a tendency towards classical or modern programming)

In contrast to the aforementioned, the evaluation of students' solution to the problem tasks is completely objective. Herein lies the uniqueness of this event, because this is the only competition where complete objectivity is achieved in the evaluation process. One can find such objectivity in sports competitions, if one ignores the possible impact of the referees.

In fact, during the evaluation, or programme testing in general, one does not examine the structure of the programme. The output data which the programme provides present the only relevant indicator of whether the task has been successfully solved or not. Partial evaluation does not exist. If the programme reports a syntax error, the task scores zero points, while the waiting time for the execution of the programmes is determined in advance.

This method of evaluation has other important features:

- it does not make students uncomfortable because nobody evaluates their way of thinking,
- it opens the possibility for creativity,
- it ensures freedom of access to problem solving,
- it encourages bluffing (frequent is the use of solutions which are generated randomly).

Students often participate as competitors in problem solving and in creating their own applications. However, it can be noted that their performance is quite different in these two categories and the reason for this difference lies in their individual characteristics.

On the other hand, a good programmer is often not a good competitor, because he or she is more prone to the development of commercial software, involving the appropriate applications and an independent choice of programming tools. The literature which one needs to study in order to prepare for the competition in computer science is equivalent to the literature studied for the competitions in other subjects and will not be discussed at this point.

THE DEVELOPMENT OF COMPUTER SCIENCE COMPETITIONS IN CROATIA

Although there were competitions in computer science held in former Yugoslavia, this paper only deals with the competitions held form the beginning of the 90s to the day. Computer science competitions in Croatia began in 1992. Over the years, the competition system has witnessed various changes which reached their climax in 2010. Following the development of computer science competitions, another important peculiarity of these events compared with the competitions in other subjects should be highlighted.

The competitions of primary and secondary school students were held at the same time and contestants were divided into two categories; all primary schools students competed in the first category, while all high school students competed in the second category. The competitions were not related to the curriculum of the subject. One can note that this kind of detachment, the separation of the competition from the content of the material taught as part of the school subject curriculum, is very rare. Therefore, an additional extra-curricular involvement of teachers and students is required. Only in 1992, one part of the competition tested the theoretical knowledge of computer science, and some primary school pupils took part in this. In this part of the competition, students were given a standard test which examined their knowledge in computer parts, input and output units, data memory and number system. This part of the competition was not particularly appealing to students, so it was discarded the following year.

During the first years of the competition, contestants were invited to the national competitions according to the results they had achieved at the municipal competitions. After 1995 these criteria were changed enabling a better selection. Students who took part in the national competition were among the best ones in the county competitions.

Already at the very beginning of computer science competitions one could also find competitions based on problem solving. In primary schools those competitions could be divided into the competitions in programming using Logo or QBasic programming languages. The competitions in the Logo programming and in QBasic programming were carried out separately, and their tasks were completely different. Students had chosen themselves which of those competitions they wanted to participate in. Additional changes were introduced in 1996 when it was allowed to switch from problem solving in the QBasic programming language to the Turbo Pascal programming language. At first, high school students could choose which of the three programming languages they would use in solving the tasks: QBasic, Turbo Pascal or C. In year 2001, the use of QBasic was excluded, since this programming language had become old fashioned. In 2004, the use of C++ was allowed, and no further changes were made.

The evolution of IT technology enabled the reduction in the time one had to wait to see the result of the programme. It was reduced from the standard 10 seconds to just one second. This was quite reasonable because the programmes were initially tested on PC-286 and PC-386 computers.

The competition in the basics of computer science was introduced in 2008, in order to motivate a greater number of students, who did not want to spend a substantial part of their spare time to prepare for the competition. This contest, which closely follows the curriculum of the high school subject, is open to all high school students. Unfortunately, all students compete in the same category. This competition will be described in more details later in the paper.

Eventually, it became clear that more categories were needed in the computer science competition. For example, the first places in the state competitions for high school students were always won by the final grades students, who, naturally, were more experienced than the first or second grade students. In addition, the advantages were also related to the students' knowledge of mathematics which was often essential in solving the given tasks.

Confronting their older colleagues was a great challenge for the younger students and the inability to achieve good results often resulted in the weakening of their motivation to participate in the competition next year. In order to overcome this problem, the results were ranked individually for each grade in 1998. Therefore, the results of the first and second grade students were separated from the results of the third and fourth grade students.

Similarly, in the primary school competition, 5-6 graders and 7-8 graders were ranked separately. Furthermore, in 2000 students were divided into age groups and were given different tasks, appropriated to their age. The final move was made in 2004 when each class was ranked separately. The age groups remained the same. This represented a decisive step towards the final popularization and acceptance of these competitions among the students of all ages.

The competitions in programming at the primary school level do not require special knowledge in algorithms and data structures. The basic prerequisites are the ability to understand a given problem, to solve it in the specific programme language and to analyze the strategy chosen for problem solving. The precise parsing and determination of the special cases that may occur in the given procedure are also required. Of course, the above requirements vary depending on the programming language used.

The software fair has not experienced any significant changes from the moment of its inception. When the local board at the county contests chooses the best applications, the national board decides which participant is to be invited to the national competition where the designed software is presented once more. The board has to check whether the contestants have made the software themselves. This is very important. In the previous years a decrease in student-made applications had been noticed, so in 2010 it was decided that Microsoft would take over the review process of students' software in 2010. This raised questions about the impartiality of the technology, but as a result of this change, the software fair witnessed a tendency toward a greater quality of the software presented.

Recently a dispute has been raised between the Education and Teacher Training Agency (AZOO) and the Croatian Association of Computer

Science (HSIN). After nearly twenty years of working in the field, HSIN withdrew from the organisation of the competitions in computer science, keeping alive the only Croatian Open Competition in Computer Science (HONI) through which they continued selecting Croatian representatives in the international competitions. This reveals another peculiarity of the competitions in computer science which distinguishes them from the competitions in other school subjects – the laureates of the national competitions do not necessarily participate at the international competitions.

COMPETITION CATEGORIES

Logo

The demands placed upon the students when programming in Logo can be divided into geometric tasks and working with lists. The geometric problems require the knowledge and understanding of Euclidean geometry. Solving geometric problems develops an impression of the size of certain angles and the position of objects in the plane and space, e.g. when drawing regular polygons, and star-like figures, it is necessary to know the relationship between the size of the angles and the number of vertices. Students who are talented in geometry have initial benefits, but continuous exercise reduces any potential differences between the contestants. The demands in solving a given problem include combining graphical solutions with the basics of programming, such as the use of loops or recursion. From the pedagogical point of view, this provides students with the best path to adopt a routine in programming and using procedures.

When working with lists, it is expected that students understand how to use sequences of similar data, their classification and examination of properties.

QBasic / Pascal

There are two types of problems contestants are presented with in this category - the problems that are solved using numerical manipulations and the problems related to working with strings, i.e. sequences of characters.

Primary school students are not expected to be familiar with complicated algorithms and data structures, but they should be able to work with functions and procedures in the programme they decide to use. They are also supposed to be familiar with files, arrays, matrices and programming loops.

Usually there are no extremely large inputs in the test-examples, but some specific (marginal) cases often occur and students have to pay attention to them when solving the problems.

Almost all problems are based on precise reading of the input and the appropriate choice of storing that data into variables, where it is very important for students to be familiar with the distinct types of variables.

Common types of problems are presented in the following short examples:

Suppose there are sequentially given authors' names and titles of their works, which should afterwards be written in a specific form, similar to the references at the end of the scientific paper. During this work with the sequences of symbols, primary school students are supposed to adopt a method of proper reading of the given input, to separate the crucial parts of the given information (the authors' names from the titles of their works), to sort given data and to print that data out in a desired way. Possible additional problems can appear in the case of repeating the authors' names.

The problems of his type help develop further manipulation with the lists of similar information, teach students to separate them according to their characteristics, and provide additional inclusion of the language-type problems into the computing procedure.

A similar problem, which combines previous approach and further studying of the elementary number theory, is the task in which students are supposed to print given numbers in words. While solving this problem, they need to determine the size of a given number and find its digits. Afterwards, the linguistic-symbolic aspect of the task is involved in the procedure. This represents a convenient cognitive model for transferring real-life situations into a computer language program. Besides that, these problems improve students' concentration in task-solving, resulting in cautiousness and precise study of a given problem.

High schools – programming

According to many experts, this is the top competition in computer science. This category is both demanding and challenging, leads to maximal competitiveness, makes contestants give their best and dictates success by providing a proper combination of talent and persistence. Because of the mentioned properties, programming competitions are the most common competitions in computer science in the world. Besides that, this category is adjusted to all high school students and each interested contestant is able to obtain a reasonable number of points, provided they apply the right approach.

The reasons for that are the structure of a given problem and the selection of test-examples. These problems are more complex and complicated than the problems given to contestants in the previously described category, so only the extraordinary algorithm, combined with the proper use of data structures, results in winning the maximum number of points. However, many algorithmic solutions, even with some deficiencies, are awarded with a certain number of points.

Obviously, the knowledge of various data structures and algorithms is needed to solve the given problems, but it is not necessary. Many algorithms, made without the use of the higher-level theoretical knowledge may also win some (fewer) points.

Take, for example, a standard type of problem of finding the longest subsequence with a given property, which consists of not necessarily consecutive elements of some sequence (for instance, the contestants may be required to determine the longest subsequence that first increases and then decreases). The most common approach of solving this problem by writing a greedy algorithm will seem appropriate in some cases, but test-examples are purposefully chosen in order to distinguish this solution from the correct one (i.e. greedy algorithm will fail in some cases).

So, for achieving top results, contestants need to be familiar with various types of algorithms, such as dynamic, generic or backtracking algorithms, which need to be additionally improved in order to allow the manipulation of a great quantity of data in a short amount of time.

Therefore, competitors should be familiar with some advanced data structures, such as lists, stacks or heaps. Apart from the procedural knowledge of the mentioned structures, contestants should have some experience with their usage, which provides them with the ability to apply their knowledge in the given time.

Such an approach to programming problems represents the natural extension of the computer science competitions for primary school students. The basic idea was to develop students' interest in computer science, giving them the opportunity to learn how to use programming languages and adopt the basics of programming. The main idea behind this process was to enable students to deal systematically with some complicated algorithms, making them capable of analyzing algorithm structures and distinguishing algorithm components, in order to improve them and apply them to the given problems (students' continual individual work is essential in this process).

The fundamental sources of the required knowledge are:

- teachers, who are supposed to have the experience, knowledge and willingness to work,
- literature available at the school libraries,
- computer science clubs, where teachers and former competitors usually volunteer,
- interaction with colleagues,
- access to problems that have appeared in the past contests, which are available on the internet.

For a certain number of contestants, placement in the competition is not very important and such students are not going to spend much time preparing for the competition. Because of the competition structure, a chance of achieving a fine result is given to almost everyone. On the other hand, many students already see programming as their future profession or think of it as a challenge or interest in their teen years. For such students, preparing and participating in the computer science competitions has additional benefits as they are allowed to learn significant parts of some undergraduate courses which are an essential part of the electrical engineering and mathematics studies, which happen to be the first higher education choice of most contestants.

Perhaps the most challenging part of programming competitions consists of problems, which have non trivial solutions, and clear ideas for solving can be partially learned using the properly chosen literature. Those are purely mathematical problems, and the solutions lie in the specific algorithms (for instance, a winning strategy for a game). Generally, problems of this type appear only at the competitions of a higher rank as is the Croatian Computer Science Olympics.

Such problems are extremely innovating and force both primary school pupils and high school students to test their own intelligence and study additional literature. It is clear that a wide range of knowledge is needed for one to achieve good results at these competitions. Therefore, these competitions expand the areas they cover.

As a matter of fact, competitors in computer science often happen to be very successful competitors at mathematics and/or physics contests. This fact provides a strong interaction of the mentioned disciplines already at primary school and high school levels. The list of all previous problems can be found on the website http://www.hsin.hr/dmih.

High school – the Basics of Computer Science

This rather young competition consists entirely of pen-and-pencil problems covering a wide range of topics in the basics of computer science, such as:

- computer architecture,
- number systems,
- logics circuits and Boolean algebra,
- networking,
- fundamentals of the algorithms (computer code analysis),
- Microsoft Office applications

This competition lasts one hour. Although, there is no big difference between the problems that appear in the competition in the following years, each student is allowed to participate only once on the national competition in this category.

There is a very close relation between the covered topics and the curriculum of the high school computer science subject. Therefore, contestants are ranked according to the type of their high schools (schools whose educational programs are based on mathematics are ranked separately from the other schools).

Because of the great variety of the covered topics and relatively strict time limits, the key to success is exceptional training and preparation, implying students' high degree of self-regulation and extreme self-control in learning.

An interesting feature of this competition lies in the fact that areas covered by given problems do not transcend the curriculum of high-school computer science, so preparing for it requires no additional engagement in terms of using the additional literature or media. The stated fact makes competing in the basics of computer science acceptable to a wider group of high school students, so it is reasonable to expect that it will expand in the forthcoming years. Besides that, it can be observed that the fields covered by this contest greatly coincide with the curriculum of the computer science, which is examined on the graduation tests. For this reason, preparing for this contest may serve as an introduction to those tests, which increases its popularity.

High school – HONI

The original idea of this contest was to prepare pupils for the national competition in programming, but its role has recently changed significantly (as was already mentioned).

HONI is organized as a league system, which consists of seven rounds which are organized in specific intervals during the whole school year (rounds are usually held on Saturdays). In each round high school students are given three to four programming problems which they solve individually. It should be pointed out that all contestants solve the same set of problems. The structure of the given problems is much the same as the structure of the problems given in the national competition in programming. All the problems which have been given at this competition can be found on the following web-page **http://www.hsin.hr/honi**.

The points won are transferred into the next round. At the very end, prizes are awarded to individual contestants and teams (all contestants compete in the same category). In team scoring, high schools are ranked by comparing average results of contestants from each school that participated at the competition.

In addition to developing a competitive spirit in some institutions (e.g. schools in smaller communities), HONI provides additional motivation for students. Because of the many rounds of this competition and the fact that it lasts throughout the entire school year, contestants are given the chance to improve their results from the earlier rounds, and have the advantage of working at their homes. Also, contestants are able to gain some extra motivation for further work by tracking their own progress through the rounds, which makes them less nervous. Besides that, according to (Vukasović, 1994:371) "proper recording of the results of work or study sustains the interest, stimulates and gives dynamics".

These facts emphasize the advantages of this competition over the national competition in computer science, which usually lasts two consecutive days and in which bad mood or pre-competing anxiety can play an important role.

Science fair of the computer software programs

This category has been mentioned already. Its great advantage is in developing teamwork as primary school pupils or high school students make a program together. Besides that, this category is more suitable for competitors which prefer not to be limited by time and are used to work only in their own home and on their own equipment, while different situations can cause them anxiety.

Thus, working on their own software gives students an opportunity to develop their programming skills. However, it can be observed that the presented programs follow certain trends – in the early 90s most programs were based on various themes related to Croatian history, geography, culture and music (those were mainly programs with an educative character). After that, there was a penetration of graphical programmes (Paint type) and finally programs dedicated to networking. This decreases creativity and originality, and provides a greater level of commerciality, i.e. it seems that students try to adjust their programmes to the demands of the market, thereby losing the real, healthy competitive character.

High schools – ACSL

The last competition that is going to be discussed here is not a fully Croatian competition. ACSL stands for the American Computer Science League. The highest interest for this competition was shown in Croatia in the late 90s, when high schools from Zagreb, Osijek, Požega, Rijeka and Pazin joined ACSL. Originally conceived as a US contest, it first spread to Canada and Mexico, and during the 90s to Europe, when teams from England, Germany, Romania and Croatia joined.

As a matter of fact, this is a very special type of contest in which an especially interesting combination of team work and individual work is involved. ACSL consists of four preliminary league rounds and the finals. The first four rounds are held at the participants' home institutions, while the finals take place at the pre-selected city in the United States. The contestants are divided into three categories, which are chosen when students apply for the competition. These categories are called Junior, Intermediate and Senior and they are roughly, but not strictly, specified by the age of the contestants.

In the first rounds high school students solve the given problems individually, and in each round they are given five theoretical problems, each worth one point, and a programming problem that is tested on five test examples, each worth one point. Types of problems are the same for each category, but differ in complexity.

After all rounds, which are held in winter and spring, the results are added together and the teams, which are to represent schools at the finals, are formed. In the Junior category each team consist of five high school students, while in the other two categories teams can have three or five members. The number of points that the team scores on the league round is the sum of all points of the team members, and according to these scores the organizers choose teams which are invited to the finals.

The number of the registered teams is usually far greater than the number of teams which participate at the finals, so a full engagement of the competitors is required from the very beginning.

At the finals, contestants individually solve six or twelve short theoretical problems, depending on their age, and they solve four or six programming problems (depending on the age and on the number of team members) working together as a team. Five-member teams have three computers at their disposal, while three-member teams have only one. The success in this part of the competition is determined by the well-organized team work and a proper distribution of the roles within the team.

In the programming problems one encounters matrix manipulations, the elementary number theory, problems related to analytic geometry, binary trees and strings. Examples of these problems can be seen at www.acsl.org.

The theoretical part (the so-called short problems) of this competition coincides almost completely with the one covered by the competition in the basics of computer science. This part of the competition seems to be partially built on the foundations of the ACSL, which has a great chance to become interesting as a supplement in the preparations for the final examinations in computer science.

The necessity of good teamwork makes a difference between this and the aforementioned contests. The theoretical approach to this type of social work has been described in literature in many places, because of the wide range of interests, which team work shows on many different levels. So, it certainly deserves to be commented on.

ASPECTS OF TEAM WORK

From the employers’ points of view, and from the requests of today’s business market, performing work assignments is not the same as it was about 20 years ago. Nowadays, the employee is required (besides the expected level of knowledge) to be motivated to study, improve his knowledge, invest in technological and cognitive education, to have a great desire to develop his own potentials, but also to have a number of social skills (Miljković et al., 2002). Since a human being adopts the mentioned social skills in the course of the educational process (Bognar, Matijević, 2002), each occasion in which these skills appear should be noted. One of these skills is the affinity for team work and the ability to fit in a team as well as any other working environment. Participating in the contests of the described type stirs interest in team work, and helps achieve the goal of preparing both primary school and high school students for carrying out their activities at a high and modern level. This approach is described in details in (Vizek-Vidović et al., 2002).

From the pedagogical point of view, team work presents one of the most prestigious social forms in the training – education process, because it increases competitiveness and causes additional motivation by all members of the team. More about this theoretical approach can be found in (Bognar, Matijević, 2002).

Finally, an especially interesting aspect of the optimization problem can be related to this sort of work. A common problem in practice involves determining the optimal strategy for solving the given problems. Since many programming contests require team work at some point, much time has been spent in studying the efficiency of different strategies, which have an educational-training character, besides the scientific one.

An extremely precise overview of the wide range of strategies, together with their advantages and disadvantages, is given in the manuscript (Trotman, Handley, 2008). It is not convenient to go into details of the sparse strategies here, but it suffices to point out that the authors of the mentioned manuscript claim that there is no single strategy that will reduce the overall time to solve all problems, and there is not even a polynomial time algorithm to find that strategy. So, upon entering the competition the team members must decide, how many problems they are likely to solve and how many problems other teams will solve.

But, at this point it is natural to agree with (Manzoor, 2001:4) and note that “success in programming contests is affected by factors other than skill, most importantly adrenaline, luck, and the problem set of the contest”.

CONCLUSION

It is rather important to point out that the competition is an environment in which various aspects of computer science can be taught (Manne, 2000). One of those aspects is definitely teamwork, which is much more than just a part of programming. Competing presents one with an extremely useful opportunity for the development of many social skills, besides the basic ones – such as cognitive skills and technological education.

Each competitor or each team “must extract the essence of the description, formulate it as a mathematical problem, and then apply robust computer science theory to solve it" (Trotman, Handley, 2008:3). An example of how this might be done is given in (Shilov, Yi, 2002). We give the concluding comments of these authors (Shilov, Yi, 2002:101):

- „Programming contests are a good opportunity for a better education and the popularization of computer science theory and mathematical foundations of the formal methods.
- Computer science journals and magazines should promote popularization of computer science theory.
- An attitude of the theoreticians to the popularization and contests should and can be improved."

Old programming contests problems could be used to teach data structures and algorithms as is suggested by (Szuecs, 2001). Also, competition is an environment in which aspects of computer science such as object-oriented design could be introduced (Andrianoff et al., 2004).

As it can be seen from the description above, in the last 15 years competitions in computer science in Croatia have reached a high level of development. One of the latest competitions is The Croatian Open Competition in Computer Science, which is an internet programming contest that is held without the necessity of being settled in a particular institution. Such type of a contest is extremely popular throughout the world and the interest in this type of a competition is increasing rapidly. For a deeper discussion we refer the reader to (Astrachan et al., 2003). Croatia follows modern trends through the HONI competition.

The following quote imposes itself as the final conclusion (Dagienè, 2009:1): "Although computer science is not taught as a discipline in many countries, pupils are invited to participate in different contests on informatics organized all over the world. When pupils get interested in programming contests, they look for training and gain some computer science education. Contests are exceptionally valuable for motivating and involving pupils in computer science".

REFERENCES

Andrianoff, S. K., Hunkins, D. R., Levine, D. B. (2004). *Adding objects to the traditional ACM programming contest,* In Proceedings of the 35th SIGCSE technical symposium on computer science education, (pp.105–109)

Astrachan, O., Khera, D. B., Kotz, D. (1993). *The Internet Programming Contest: A Report and Philosophy*, ACM SIGCSE Bulletin, *25*(1),(pp. 48–52)

Bognar, L ., Matijević, M. (2002). *Didaktika*, Školska knjiga, Zagreb

Dagienè, V. (2009). *Sustaining Informatics Education by Contests, Lecture Notes In Computer Science,* Proceedings of the 4th International Conference on Informatics in Secondary Schools - Evolution and Perspectives: Teaching Fundamentals Concepts of Informatics, *5941*, (pp. 1–12)

Kleitzen, B., Vizek-Vidović, V., S., Cota Bekavac, M. (2002). *Aktivno učenje i ERR okvir za poučavanje – priručnik za nastavnike*, Zagreb: Forum za slobodu odgoja

Manne, F. (2000). *Competing in computing*, Proceedings of the 2000 Norsk Informatikkonferanse, (pp. 129–138)

Manzoor, S. (2001). *Common mistakes in online and real-time contests*, Crossroads, *7*(5), (pp. 4)

Miljković, D, Rijavec, M., Vizek-Vidović, V., Vlahović Štetić, V. (2003). *Psihologija obrazovanja*, IEP-VERN, Zagreb

Shilov,N.V.,Yi,K.(2002).*Engaging students with theory through ACM collegiate programming contests*, Communications of the ACM, *45*(9), (pp. 98–101)

Szuecs, L. (2001). *My favorite programming contest problems*, Journal of Computing Sciences in Colleges, *17*(1), (pp. 225–232)

Trotman, A., Handley, C. (2008). *Programming contest strategy*, Computers & Education, *50*, (pp. 821–837)

Vukasović, A. (1994). *Pedagogija*, Alfa d.d. Hrvatski katolički zbor “Mi”, Zagreb

Ljerka Jukić
Department of mathematic, Josip Juraj Strossmayer University of Osijek
Trg Ljudevita Gaja 6, 31 000 Osijek, Croatia
ljukic@mathos.hr

Ivan Matić
Department of mathematics, Josip Juraj Strossmayer University of Osijek
Trg Ljudevita Gaja 6, 31 000 Osijek, Croatia
imatic@mathos.hr

ULOGA NATJECANJA U KURIKULU NASTAVE INFOMATIKE

SAŽETAK

U posljednje vrijeme natjecanja iz školskih predmeta polako ali sigurno zauzimaju istaknuto mjesto u odgojno-obrazovnom procesu te imaju značajnu ulogu pri motivaciji i inspiriranju učenika za pojačan daljnji rad. U ovom radu je dan pregled natjecanja iz informatike u Hrvatskoj s posebnim naglaskom na njihove prednosti i nedostatke te uz isticanje njihove važnosti i primjene u metodici nastave informatike. Također su prikazane te primjerima potkrijepljene pozitivne posljedice sudjelovanja učenika na natjecanjima, poput pojačanog razvoja socijalnih i spoznajnih vještina.

Ključne riječi: *metodika nastave informatike, natjecanja iz informatike, razvoj socijalnih vještina.*

UVOD

Natjecanja se već duže vrijeme ističu kao važna odgojna sredstva u pedagogiji i odgojnom radu. Natjecanje je „oblik aktivnosti u kojoj se pojedinci ili grupe ljudi zalažu brže i kvalitetnije ostvariti zadatak od drugih pojedinaca ili grupa kojima je povjeren isti zadatak, te time djeluje kao pokretačko sredstvo i potiče na veće zalaganje (...) Natjecanje kao sredstvo odgoja ima svoj smisao. Ono djeluje na osjećaje, unosi među natjecatelje određenu živost i vedrinu, razvija zanimanje, budi borbeni duh, jača volju i želju za pobjedom. Time ono potiče na uspjeh, na akciju, tražeći da se radni zadatak obavi na vrijeme, što je prije moguće i kvalitetno.“ (Vukasović, 1994., str. 370)

Praćenjem razvoja tehnologije te infiltriranjem računala u svakodnevni život i većinu kućanstava unutar posljednjih 15-ak godina u hrvatskom školstvu se jednakim tempom razvijaju i natjecanja iz informatike. Naravno, jednak razvoj i popularnost natjecanja iz tog predmeta je prisutan i u drugim državama, kako u visoko razvijenim, tako i u onima nešto slabijeg stupnja razvoja. U ovom radu će se moći primijetiti kako u raznovrsnosti i rezultatima hrvatski učenici nimalo ne zaostaju za kolegama u drugim državama.

U Hrvatskoj se prva natjecanja iz informatike pojavljuju početkom 90-ih godina prošlog stoljeća, kada su organizaciji svakog od tih natjecanja prethodili brojni problemi. Prvenstveno su ti problemi bili tehničke prirode, jer je tada u mnogim dijelovima države bio pravi izazov formirati mjesto s dovoljnim brojem računala na kojima će učenici raditi. Osim toga, u to vrijeme je posjedovanje osobnog računala bila razmjerno rijetka pojava te je iz tog razloga bilo potrebno pokušati unaprijed osigurati i opremu potrebnu za pripremu za natjecanje.

Slične potrebe za posebnom opremom su prisutne i u drugim predmetima, kao npr. u kemiji ili fizici. No, opremljenost škola potrebnim materijalima i nastavnim sadržajima za te predmete je prisutna već dulje, čak i unazad 40-ak godina.

Nasuprot tome, 1992. i 1993. informatika se tek polako i sramežljivo uvlačila kao punopravan predmet u školski program, pa su tek rijetke škole posjedovale računalnu opremu. Radi toga se većina informatičkih natjecanja u tim ranim danima održavala u specijaliziranim klubovima i zajednicama tehničke kulture, koji su na raspolaganju imali dovoljan broj relativno pogodnih računala. Od samih početaka je bilo jasno kako oprema na kojoj učenici rade treba biti što sličnija te se inzistiralo na tome da učenici rade na osobnim računalima, a ne na računalima marke Spectrum, Atari, Commodore, Amiga i sličnima.

Shodno tome, sve do druge polovice 90-ih natjecanja iz informatike u Osijeku i Osječko-Baranjskoj županiji održavala su se istodobno na nekoliko mjesta - Zajednici tehničke kulture, III. gimnaziji i Elektrotehničkoj školi.

Razvojem računalne opreme, brojnim donacijama tijekom 90-ih te pokazivanjem povećenog interesa učenika za informatiku ovi su početni problemi uspješno prebrođeni. Također, posjedovanje vlastitog računala je postalo i potreba, a prestalo je i biti luksuz te ih danas ima većina učenika.

No, povećao se i broj natjecatelja, pa se i danas natjecanja održavaju na više lokacija, jer se inzistira na istodobnom pristupu, što smanjuje mogućnost curenja podataka.

Posebnost natjecanja iz informatike leži u tome što učenicima pružaju mogućnost da odaberu pristup koji im više odgovara:

- Rješavanje problemskih zadataka - učenici (uglavnom pojedinačno) rješavaju unaprijed sastavljene problemske zadatke, gdje se od njih traži da izrade programe koji će raditi što brže i preciznije.
- Smotra softverskih radova - učenici sami (dopušten je i ekipni rad) izrađuju program koji na natjecanju prezentiraju unaprijed sastavljenoj stručnoj komisiji. Pri tome im se unaprijed ne zadaju nikakvi propisi, sami odabiru temu i programski jezik u kojem će

izrađivati program. Za razliku od prvog pristupa, ovaj dopušta krajnji razvoj učeničke kreativnosti te ih ne ograničava vremenom predviđenim za izradu programa.

Prisutnost komisije koja daje ocjenu prezentiranog softverskog rada automatski pridonosi subjektivnosti vrednovanja. Kako bi se smanjila subjektivnost, postoje unaprijed određeni kriteriji ocjenjivanja, koji su poznati i samim natjecateljima.

Neki kriteriji su:

- originalnost ideje (npr. radi li se o poboljšanju postojećeg programa, je li program korisničke ili edukativne prirode),
- kreativnost izvedbe i prilagođenost korisniku,
- moguće pogrješke ili nesavršenosti u radu programa (pri radu s velikim brojem podataka ili radu na slabijem računalu),
- programsko rješenje (korištenje algoritamskog znanja, jednostavnih ili složenijih rješenja, tendencija prema klasičnom ili modernijem programiranju).

Nasuprot prethodnome, vrjednovanje učeničkih rješenja problemskih zadataka je potpuno objektivno. U tome leži i posebnost ovih natjecanja, jer predstavljaju jedina natjecanja gdje je postignuta potpuna objektivnost pri ispravljanju zadataka, osim možda u sportskim natjecanjima ako se zanemari mogući utjecaj sudaca.

Naime, prilikom ispravljanja zadataka, tj. testiranja programa, uopće se ne gleda kôd programa ni način na koji je učenik rješavao zadatak. Jedino što je relevatno pri ispravljanju su izlazni podatci koje program daje. Djelomičnog ocjenjivanja nema, ukoliko program javi sintaktičku pogrješku boduje se s nula bodova, dok je vrijeme čekanja izvršenja programa propisano unaprijed.

Ovakav način ocjenjivanja ima i druge značajke:

- ne stvara učenicima pritisak da će netko procjenjivati njihov način razmišljanja,
- otvara mogućnost vlastite kreativnosti,
- osigurava slobodu pristupa,
- može potaknuti i blefiranje (česta pojava je korištenje generiranja slučajno odabranog rješenja).

Nerijetko se pojedini učenici okušaju kao natjecatelji i u rješavanju problemskih zadataka i u izradi vlastitog softvera. No, može se primijetiti kako se njihova uspješnost u ove dvije kategorije prilično razlikuje, a razlog tome su njihove individualne karakteristike. Osim toga, dobar programer često nije i dobar natjecatelj, jer je skloniji izradi komercijalnog softvera po narudžbi, uz prikladne aplikacije i samostalan odabir programskog alata.

Nužnost literature potrebne prilikom pripreme za natjecanja iz informatike je jednaka onoj potrebnoj za natjecanja iz drugih predmeta pa tome na ovom mjestu ne će biti posvećena posebna pažnja.

RAZVOJ NATJECANJA U HRVATSKOJ

Iako su se natjecanja iz informatike održavala još u vrijeme Jugoslavije, u ovom radu razmatraju se natjecanja održana od početka 90-ih. Natjecanja iz informatike u Hrvatskoj započinju 1992. Tijekom godina je sustav natjecanja doživio različite izmjene, koje su svoj vrhunac doživjele upravo ove godine. Praćenjem njihova razvoja usput se ističe još jedna bitna posebnost ovih natjecanja s obzirom na natjecanja iz drugih predmeta. Istodobno se održavaju natjecanja učenika osnovnih i srednjih škola, koji su podijeljeni u dvije kategorije. Dakle, u istoj kategoriji su svi učenici osnovnih škola te se u istoj kategoriji natječu i svi učenici srednjih škola. Očito, ova natjecanja nisu izravno vezana uz školsko gradivo iz predmeta informatika. Primjetno je kako je nevezanost natjecanja uz sadržaj nastavnog gradiva vrlo rijetka pojava, koja automatski zahtijeva veći dodatni izvannastavni angažman i profesora i učenika.

Treba napomenuti kako su prvih godina natjecatelji na državno natjecanje pozivani na osnovi rezultata općinskih, a kasnijih godina (nakon 1995.) na osnovi rezultata županijskih natjecanja. Učenici koji ostvare najbolje rezultate na državnom natjecanju dobivaju poziv za sudjelovanje na međunarodnim natjecanjima.

Prve se godine održalo i natjecanje iz teorijskog znanja informatike, u kojem se natjecao dio učenika osnovnih škola. Na tom dijelu natjecanja učenici su ispunjavali standardni test o poznavanju dijelova računala, ulaznim i izlaznim jedinicama, memorijskim podatcima te brojevnim sustavima. Kako se ovo natjecanje nije pokazalo posebno primamljivo učenicima, ukinuto je već iduće godine.

Osim tog dijela natjecanja od samih početaka su u osnovnoj školi prisutna i natjecanja u rješavanju zadataka problemskog tipa. Dijele se na natjecanja u programiranju u programskom jeziku Logo te u programskom jeziku QBasic. Preciznije, natjecanja u programiranju u Logou i u QBasicu su odvojena, s potpuno različitim zadatcima te učenici sami biraju na koja od tih natjecanja žele ići. Dodatna izmjena je uvedena 1996. godine, od kada je učenicima dopušteno da umjesto u QBasicu zadatke rješavaju i u programskom jeziku Turbo Pascal.

Srednjoškolci su ispočetka mogli zadatke rješavati u programskim jezicima QBasic, Turbo Pascal ili C. God. 2001. je isključena mogućnost korištenja QBasica, a 2004. je dopuštena i upotreba programskog jezika C++. Daljnjih izmjena nije bilo. Evolucijom računala se nekadašnje vrijeme čekanja na rezultat programa, sa standardnih 10 sekundi, smanjilo na samo jednu sekundu. No, to je posve razumno jer su ispočetka programi bili testirani na računalima tipa PC-286 ili 386.

Kako bi se dodatno motivirao za sudjelovanje što veći broj učenika, koji možda i nisu spremni odvojiti znatan dio svog slobodnog vremena za pripremu za natjecanja, 2008. uvedeno je dodatno natjecanje u konkurenciji učenika srednjih škola - natjecanje iz osnova informatike. Ovom natjecanju, koje usko prati gradivo srednjoškolskog predmeta, mogu pristupiti svi učenici srednjih škola, no nažalost se svi natječu u istoj kategoriji. Nešto kasnije će ovo natjecanje biti temeljitije opisano.

S vremenom je postalo jasno kako na neki način treba doskočiti malom broju kategorija u ovim natjecanjima. Primjera radi, prva mjesta na državnim natjecanjima učenika srednjih škola su odnosili učenici 3. i 4. razreda, koji su ipak imali više iskustva od učenika nižih razreda. Osim toga, nalazili su se i u prednosti s obzirom na gradivo koje su slušali tijekom nastave, ne samo iz informatike, već i iz matematike, što je često pri rješavanju zadataka od presudne koristi.

Iako je mlađim učenicima veliki izazov i dodatni poticaj sučeljavanje sa starijim kolegama, nemogućnost postizanja posebno dobrih rezultata i ostvarivanja plasmana adekvatnog uloženom trudu često rezultiraju slabljenjem motivacije. Kako bi se ovaj problem premostio, najprije se 1998. uvode zasebne liste te su učenici 1. i 2. razreda rangirani zasebno od učenika 3. i 4. razreda. Slično tome, u osnovnoškolskoj konkurenciji se prave zasebne liste učenika 5. i 6. te 7. i 8. razreda.

Nadalje, od 2000. godine učenicima se, podijeljenima u starosne skupine, počinju davati različiti zadatci, primjereniji njihovoj dobi, a konačni je pomak učinjen 2004. godine kada se počinju izrađivati zasebne liste za svaki razred, dok starosne skupine za dodjelu zadataka ostaju iste. Time je učinjen presudni korak prema krajnjoj popularizaciji i prihvatljivosti ovih natjecanja učenicima svih dobnih skupina.

Natjecanja iz programiranja na osnovnoškolskoj razini od učenika ne traže posebno poznavanje tipova algoritama ni baratanje s različitim strukturama podataka. Osnovne postavke od kojih ova natjecanja polaze su poznavanje korištenog programskog jezika, razumijevanje zadanog problema te sposobnost temeljite analize u pristupu rješavanju. Također se traži i precizno raščlanjivanje i određivanje specijalnih slučajeva koji se u danom postupku mogu pojaviti. Naravno, navedeni zahtjevi se razlikuju ovisno o programskom jeziku koji se koristi.

Smotra softverskih radova nije doživjela bitne izmjene od samih početaka. Nakon što lokalne komisije na županijskim natjecanjima odaberu najbolje radove, državna komisija procjenjuje koji od njih zaslužuju poziv na državno natjecanje, gdje se pristupa ponovnoj prezentaciji izrađenog softvera. Bitna stavka komisija je i da provjere je li prezentirani softver zaista djelo isključivo samih natjecatelja.

Nakon što je posljednjih godina primjećen pad kvalitete prezentiranih radova, ove godine dio brige oko smotre preuzima Microsoft. Nakon toga se počinju povlačiti pitanja o tehnološkoj nepristranosti, no pokazuje se tendencija prema većoj kvaliteti prezentiranih radova.

Nedavno je došlo do raskola između Agencije za odgoj i obrazovanje (AZOO) i Hrvatskog saveza informatičara (HSIN). Time se HSIN nakon gotovo 20 godina povlači iz organizacije natjecanja iz informatike, održavajući na životu jedino *Hrvatsko otvoreno natjecanje u informatici* (HONI) (uz velike poteškoće), no putem kojega se i dalje određuju predstavnici Hrvatske na međunarodnim natjecanjima. Time dolazi do još jedne specifičnosti ovih natjecanja - državno natjecanje nije izlučno za odlazak na međunarodna natjecanja.

KATEGORIJE NATJECANJA

Logo

Zahtjevi postavljeni pred učenike prilikom programiranja u Logu mogu se podijeliti na geometrijske zadatke i rad s listama. Geometrijski problemi zahtijevaju od učenika poznavanje i razumijevanje planimetrije, vrsno baratanje s udaljenostima (metričkim sustavom), a njihovim rješavanjem razvija se dojam o veličini pojedinih kutova, položaju objekata u ravnini i prostoru. Npr. za crtanje pravilnih poligona, ali i zvjezdastih likova, nužno je poznavanje odnosa veličine kutova i broja vrhova. U početnoj prednosti se nalaze učenici s izraženijim talentom za geometriju, no vježbom se te razlike mogu umanjiti do neznatnih veličina. Od učenika se traži i kombiniranje traženih grafičkih rješenja s osnovama programiranja, kao što su korištenje petlji ili rekurzija. Time se učenicima pruža konkretan i opipljiv način za usvajanje programerskih rutina i procedura.

Prilikom rada s listama očekuje se (i razvija) razumijevanje rada s nizovima sličnih podataka i ispitivanje njihovih svojstava.

QBasic / Pascal

U ovoj kategoriji prisutni su zadatci koji se rješavaju pomoću numeričkih manipulacija te zadatci povezani s radom s nizovima znakova. Od učenika se ne zahtijeva poznavanje posebnih algoritamskih postupaka i struktura podataka, no nužno je dobro baratanje s funkcijama i procedurama odabranog programa, poznavanje rada s datotekama, matricama i programskim petljama.

Obično se u test-primjerima ne radi o izrazito velikim unosima, no gotovo uvijek su zastupljeni specijalni (rubni) slučajevi na koje bi učenici pri dobrom razumijevanju problema trebali obratiti pažnju. Gotovo svi zadatci se prije svega baziraju na preciznom učitavanju podataka te njihovom zapisu u varijable, pri čemu je važno da se učenici znaju koristiti različitim tipovima podataka.

Standardni oblici zadataka su pokazani na idućim kratkim primjerima:

Neka se redom učitavaju imena pisaca i nazivi njihovih djela, koja zatim treba ispisati u određenom formatu, sličnom referencama na kraju znanstvenog rada. U tom radu s nizovima simbola učenici trebaju poznavati način pravilnog učitavanja podataka, razdvajanja bitnih dijelova (u ovom slučaju imena autora od naslova djela), sortiranja unesenih podataka te njihova ispisivanja na traženi način. Mogući dodatni problemi nastaju pri ponavljanju imena autora.

Ovakvim tipom zadataka usvaja se manipulacija nizom sličnih informacija te dodatno razrađuju postupci njihova razdvajanja i pojačanog uključivanja jezične problematike u računarstveni postupak.

Sličan problem, koji objedinjuje prethodni pristup s dodatnim razumijevanjem osnova teorije brojeva, je zadatak u kojem učenici trebaju unesene prirodne brojeve ispisati riječima. Pri tome najprije rastavljaju broj na znamenke i određuju njegovu veličinu, a zatim se u igru upliće i jezično-znakovni dio. Ovo predstavlja prikladan kognitivni model za učenje prenošenja situacija iz stvarnog života u problem dan jezikom računala. Osim toga, uvijek se dodatno vježba učenička koncentracija, oprez i precizno čitanje danih zadataka.

Srednje škole – programiranje

Prema mišljenju mnogih, ova kategorija ipak predstavlja krunu natjecateljskog programa. Istodobno zahtjevna i izazovna, sposobna razviti do vrhunca osjećaj za kompetenciju, izvući maksimum iz natjecatelja te diktirati uspjeh pravilnom kombinacijom talenta i upornosti, ova kategorija je najzastupljenija i u čitavom svijetu. Uz sve to, ova kategorija je prilagođena svima, svaki zainteresirani učenik može skupiti prihvatljiv broj bodova, ako se dovoljno potrudi te ako ima pravi pristup.

Razlog tome leži u strukturi zadataka i odabiru test-primjera. U ovoj kategoriji su zadatci daleko teži i ozbiljniji od onih u prethodno opisanoj te će samo vrhunski odabrani algoritamski postupak, kombiniran s pravilnom upotrebom strukuture podataka, rezultirati maksimalnim brojem bodova. No, mnoga algoritamska rješenja koja pokazuju neke manjkavosti, će također biti nagrađena određenim brojem bodova.

Za rješavanje zadataka potrebno je poznavanje i učenje algoritama i struktura podataka, ali nije nužno. Mnogi algoritamski postupci, napravljeni bez posebnog učenja će također rezultirati nekim brojem bodova (iako znatno manjim).

Uzmimo za primjer standardni tip zadataka pronalaženja najduljeg podniza s nekim svojstvom, koji se sastoji od ne nužno uzastopnih elemenata nekog niza (npr. ako u danom nizu proizvoljne duljine treba odrediti najdulji podniz koji raste pa potom pada). Uobičajeni pristup rješavanja pohlepnim algoritmom će proći u nekim slučajevima, no test-podatci su namjerno odabrani tako da ne prolazi u svima.

Zato se od učenika za vrhunski rezultat očekuje poznavanje različitih algoritama, kao što su dinamički, generički ili ’backtracking’ algoritmi, koje je potrebno i dodatno razraditi kako bi mogli manipulirati s velikom količinom danih podataka u vrlo kratkom vremenu.

Radi toga, natjecatelji trebaju poznavati i druge strukture podataka, kao što su liste, stogovi i hrpe. Osim samog poznavanja tih struktura, potrebno je i određeno iskustvo u njihovu korištenju, kako bi ih se moglo uspješno primijeniti u propisanom vremenu.

Takav pristup natjecateljskim zadatcima predstavlja prirodni nastavak osnovnoškolskih natjecanja, tijekom kojih je osnovna ideja bila da učenici nauče koristiti programske jezike i usvoje osnove programiranja. Nakon toga, zamišljeno je da kontinuiranim radom svladavaju kompliciranije algoritme, znaju ih analizirati, redom raščlanjivati njihove sastavne dijelove kako bi ih mogli poboljšati te i primjenjivati.

Osnovni izvori potrebnog znanja su:

1. nastavnici, koji moraju imati iskustva, znanja i volje za radom,
2. literatura dostupna u školskim knjižnicama,
3. informatički klubovi, u kojima obično dobrovoljno rade profesori i nekadašnji natjecatelji,
4. interakcija s kolegama,
5. zadatci s prijašnjih natjecanja, dostupni na internetu.

Određenom broju natjecatelja nije posebno bitan plasman pa oni ne će odvajati mnogo vremena za pripremu. No, zbog strukture natjecanja i njima je omogućeno da ostvare solidan plasman. S druge strane, mnogi već u ranijoj dobi programiranje vide kao svoj poziv, izazov ili interes pa učenjem za natjecanje zapravo ostvaruju dodatnu korist.

Može se primijetiti da učenici imaju priliku unaprijed ovladati i gradivom nekih fakultetskih kolegija, kakvi se mogu naći na nižim godinama studija matematike i elektrotehnike, a navedeni studiji su ovim natjecateljima obično prvi izbor.

No, možda najizazovniji dio natjecanja čine zadatci čija rješenja predstavljaju netrivijalne ali jasne ideje, koje se mogu dijelom naučiti pravilnim izborom literature. Radi se o zadatcima čisto matematičkog tipa, za čije rješenje je potrebno konstruirati specifičan algoritamski postupak (npr. pobjedničku strategiju za neku igru). Obično se takvi zadatci pojavljuju tek na višim rangovima natjecanja, poput Hrvatske informatičke olimpijade.

Zadatci ovog tipa su posebno inovativni pa učenike vode k testiranju vlastite inteligencije i proučavanju dodatne literature te idu u prilog činjenici da za uspješnost na natjecanju treba imati što opširnije znanje, čime ova natjecanja vidno šire pokrivena područja. Zaista, često se natjecatelji iz informatike pokazuju kao vrlo uspješni natjecatelji i iz matematike, a ponekad i iz fizike. Ova činjenica svakako ide u prilog snažnoj interakciji navedenih disciplina već na školskoj razini. Pregled svih dosad postavljenih zadataka može se pronaći na internet stranicama http://www.hsin.hr/dmih.

Srednje škole - osnove informatike

Natjecanje se sastoji iz rješavanja pismenog dijela koji sadrži zadatke iz osnova informatike, kao što su:

- građa računala,
- brojevni sustavi,
- logički sklopovi,
- mreže računala,
- osnove algoritamskog rješavanja zadataka (analiza koda),
- uredske aplikacije (alati Microsoft Officea).

Natjecanje traje jedan sat. Kako se zadatci iz godine u godinu ne mogu bitno razlikovati, uvedeno je pravilo da učenici smiju samo jednom nastupiti na državnom natjecanju u ovoj kategoriji.

Radi direktne povezanosti ovog gradiva s nastavnim gradivom, učenici su rangirani prema vrstama škole (zasebno prirodoslovno-matematičke gimnazije, zatim ostale gimnazije te zasebno tehničke škole).

Zbog raznovrsnosti zastupljenog gradiva te relativno strogog vremenskog ograničenja, za uspjeh u ovom natjecanju je potrebna izuzetna uvježbanost i pripremljenost, koja od učenika zahtjeva visok stupanj samoregulacije i kontrole prilikom učenja.

Velika prednost ovog natjecanja leži u tome što ne traži od učenika poznavanje vannastavna gradiva te time od njih ne iziskuje izniman angažman u vidu dodatne literature ili medija tijekom pripreme. Ovim natjecanje postaje prihvatljivo daleko većem krugu učenika te je za očekivati dodatnu ekspanziju ovog natjecanja idućih godina.

Osim toga, može se primijetiti kako se područja pokrivena ovim natjecanjem velikim dijelom podudaraju s nastavnim gradivom znanje kojeg se ispituje na državnoj maturi iz informatike. Time priprema za ovo natjecanje može služiti kao uvod u državnu maturu, što daje dodatan bonus njegovoj popularnosti.

Srednje škole – HONI

Prvotna ideja ovog natjecanja je bila da bude priprema za natjecanje u programiranju učenika srednjih škola, no danas je njegova uloga bitno drugačija (kako smo već i spominjali). Natjecanje se održava prema tzv. ligaškom sustavu i sastoji se od 7 kola, koja se održavaju tijekom čitave školske godine, u pravilnim razmacima (izvan nastave, obično subotom). U svakom kolu učenici dobivaju 3 ili 4 zadatka (svi rješavaju iste zadatke) koja rješavaju samostalno. Struktura zadataka je jednaka strukturi zadataka sa srednjoškolskih natjecanja u programiranju. Svi zadatci s ovog natjecanja su dostupni na stranici http://www.hsin.hr/honi.

Ostvareni bodovi prenose se iz kola u kolo. Na kraju se nagrade dodjeljuju pojedinačno, ali i timski prema prosječnim rezultatima učenika pojedinih škola.

Osim što HONI služi za razvoj natjecateljskog duha u institucijama kao što su manje škole, ono pruža učenicima i dodatnu motivaciju. Naime, zbog velikog broja kola ovog natjecanja i protezanja kroz čitavu godinu, učenici imaju priliku popraviti loš rezultat iz ranijih kola, imaju prednost rada u okruženju vlastite institucije na koju su navikli te dodatnu motivaciju za daljnji rad mogu ostvariti i promatrajući svoj napredak kroz natjecanje, što i razbija njihovu eventualnu tremu. Nadalje,"redovna evidencija rezultata rada ili učenja održava interes, unosi živost i poticaj" (Vukasović, 1994., str. 371).

Navedene činjenice predstavljaju bitnu prednost ovog natjecanja pred državnim natjecanjem, gdje se čitavo natjecanje odradi u uzastopna dva dana, pri čemu i loše raspoloženje ili trema znaju odigrati važnu ulogu.

Smotra softverskih radova

O ovoj kategoriji natjecanja već je i ranije bilo dosta govora. Velike prednosti leže u tome što se razvija sklonost timskom radu, u slučaju da više učenika surađuje u izradi programa. Osim toga, ova kategorija više odgovara natjecateljima koji ne žele biti ograničeni vremenom te su navikli raditi isključivo u vlastitom prostoru i na vlastitoj opremi, a druge situacije mogu izazvati anksioznost.

Izradom vlastitih programa im je ipak omogućeno da razviju svoj programerski potencijal. No, može se primijetiti kako su izrađeni programi ipak dijelom trendovski - početkom 90-ih je većina programa bila na temu hrvatske povijesti, zemljopisa, kulture ili glazbe (uglavnom se radilo o programima edukativnog karaktera). Zatim je došlo do prodora grafičkih programa (tipa Painta) te konačno do programa posvećenih mrežnom radu. Time se kreativnost i originalnost znatno smanjuje, dok je prisutna i sve veća doza komercijalnosti, tj. učenici teže prilagođavanju svojih programa zahtjevima tržišta, čime se gubi pravi, zdravi natjecateljski karakter.

Srednje škole – ACSL

Posljednje natjecanje koje ćemo biti opisano nije u potpunosti hrvatsko. Ovo natjecanje je vrhunac interesa na našim prostorima doživjelo u drugoj polovici 90-ih, kada su se u njega uključile škole iz Zagreba, Osijeka, Požege, Rijeke i Pazina. Kratica ACSL znači American Computer Science League, što već i samo objašnjava ovo natjecanje. Ispočetka zamišljeno kao natjecanje u Sjedinjenim Američkim Državama, najprije se proširilo na Kanadu, Meksiko, a 90-ih ina Europu, kada se natjecanju pridružuju ekipe iz Engleske, Njemačke, Rumunjske i Hrvatske. Zaista, radi se o natjecanju u kojem je osmišljena interesantna kombinacija ekipnog i individualnog rada, koju će biti opisano u daljnjem. Natjecanje se sastoji od 4 ligaška kruga i finalnog (završnog) natjecanja. Prva 4 kruga se održavaju na matičnim institucijama, dok se završno natjecanje održava u ranije odabranom gradu u SAD-u.

U samoj prijavi za natjecanje je potrebno predvidjeti u kojoj kategoriji će se koji učenik natjecati. Kategorije Junior, Intermediate i Senior su grubo, ali ne i isključivo, određeno pomoću dobi učenika. Tijekom prvih krugova učenici rade individualno, a u svakom krugu se rješavaju teorijski zadatci (koji nose najviše 5 bodova) te programski zadatak (koji se testira na 5 unaprijed određenih primjera, od kojih svaki vrijedi po jedan bod). Tipovi zadataka su isti za svaku kategoriju, no razlikuju se po težini. Nakon što se tijekom zime i proljeća održe sva 4 kruga, zbrajaju se rezultati te oforme ekipe koje bi trebale predstavljati škole na finalnom natjecanju (ekipe mogu imati 3 ili 5 članova). Broj bodova koje je ekipa ostvarila na prednatjecanjima se dobiva kao suma bodova njenih članova, a prema tim bodovima organizatori prave rang listu ekipa kojima zatim šalju poziv za završno natjecanje.

Broj prijavljenih ekipa je obično daleko veći od broja ekipa pozvanih na završno natjecanje, čime se od učenika zahtjeva pun angažman od samog početka.

Na završnom natjecanju učenici individualno rješavaju 6 ili 12 teorijskih zadataka (ovisno o uzrastu), dok se programski zadatci (njih 4 ili 6, ovisno o uzrastu i broju članova) rješavaju ekipno. Ekipe od 5 članova imaju na raspolaganju 3 računala, dok ekipe od 3 člana rade za samo jednim. Uspjeh na ovom djelu natjecanja je uvjetovan dobrim timskim radom i pravilnom podjelom uloga unutar ekipe.

U programskim zadatcima su zastupljeni rad s matricama, svojstvima cijelih brojeva, problemima iz analitičke geometrije, binarnim stablima i stringovima (nizovima znakova) te se mogu vidjeti na www.acsl.org.

Gradivo koje se pojavljuje u teorijskom dijelu se gotovo potpuno podudara s onim pokrivenim natjecanjem iz osnova informatike, za koje se stječe dojam kako je nastalo upravo na temeljima ovog natjecanja, koje bi danas moglo postati interesantno i tijekom priprema za državnu maturu iz informatike.

Upravo nužnost kvalitetnog timskog rada predstavlja razliku ovog natjecanja pred ranije opisanim. Teorijski pristup ovakvom načinu rada je opisan na mnogim mjestima, upravo radi interesantnosti koju timski rad pruža na brojnim razinama. Time svakako zaslužuje da ga se na ovom mjestu i dodatno dotakne.

ASPEKTI TIMSKOG RADA

Gledano od strane poslodavaca i zahtjeva današnjeg tržišta, obavljati radne zadaće nije više isto što prije 20-ak godina. Danas se od zaposlenika osim očekivane razine znanja zahtjevaju i daljnja želja za učenjem, napredovanjem u znanju, ulaganjem u tehnološko i spoznajno obrazovanje, želja za razvojem vlastitih potencijala, ali i brojne socijalne vještine (Miljković et al., 2002). Kako su navedene socijalne vještine dio koji ljudska jedinka usvaja tijekom odgojnog procesa (Bognar, Matijević, 2002), treba istaknuti svako mjesto na kojem one dolaze do izražaja. Jedna od tih vještina je i sklonost timskom radu te sposobnost uklapanja u tim ili radnu sredinu. Pobuđivanjem interesa za timski rad, čemu sudjelovanje na natjecanju tipa ACSL svakako ide u prilog, ostvaruje se cilj pripremanja učenika (ili studenta) za obavljanje svoje djelatnosti na visokom i modernom nivou. Ovakav pristup je podrobnije opisan u (Vizek-Vidović et al. 2002).

S pedagoške strane gledanja, ekipni ili grupni rad predstavlja jedan od najprestižnijih socijalnih oblika u odgojno-obrazovnom procesu, jer zaoštrava kompetitivnost na pozitivan način te uzrokuje dodatnu motivaciju kod svih članova skupine. Više o ovom teorijskom pristupu se također može naći u (Bognar, Matijević, 2002).

Napokon, ovom načinu radu se može pridružiti i zanimljiv aspekt optimizacijske problematike. Naime, u praksi je čest problem određivanja optimalne strategije za ispunjavanje određenih zadataka. Brojna natjecanja iz informatike u nekom trenutku zahtjevaju timski rad te je mnogo vremena provedeno u proučavanju efikasnosti određenih strategija, koje osim znanstvenog imaju i primjenjivi i odgojno-obrazovni karakter.

Posebno precizan pregled različitih strategija, zajedno s njihovim prednostima i manama, iznesen je u radu (Trotman, Handley, 2008). Na ovom mjestu nije pogodno ulaziti u detalje razrađenih strategija, no važno je napomenuti da autora navedenog rada zaključuju kako ipak ne postoji strategija koja će reducirati ukupno vrijeme potrebno za rješavanje svih danih problema te je polazni zadatak ekipe da pokuša odrediti svoj doseg i procijeni doseg drugih timova.

Ipak, na ovom mjestu se treba složiti s (Manzoor, 2001, str. 4) te konstatirati da je „uspjeh na natjecanjima iz informatike uvjetovan i čimbenicima različitim od vještine, od kojih su najvažniji adrenalin, sreća te same problemske postavke natjecanja“.

ZAKLJUČAK

Potrebno je istaknuti kako je, prema (Manne, 2000), natjecanje okruženje u kojem se mogu poučavati brojni aspekti informatike, među koje svakako pripada i timski rad, koji nadilazi aspekte same informatike. Time sudjelovanje na ovim natjecanjima omogućuje i daljnji razvoj socijalnih vještina, osim osnovnih - kao što su spoznajne vještine i tehnološko obrazovanje.

Svaki sudionik, ili svaki tim, mora, prema (Trotman, Handley, 2008, str. 3), „iz danog zadatka izvući osnovnu problematiku, formulirati ju u obliku matematičkog problema te zatim primijenitu robusnu informatičku teoriju kako bi taj problem riješili". Primjer postupka realizacije ovog procesa dan je u radu (Shilov, Yi, 2002). U nastavku su izdvojeni završni komentari tih autora (Shilov, Yi, 2002, str. 101):

- „Natjecanja u programiranju su izvrsna prigoda za bolju edukaciju i popularizaciju informatike te matematičkih utemeljenja formalnih metoda.
- Informatički časopisi bi trebali promovirati popularizaciju teorijskog pristupa programiranju.
- Stav teoretičara o popularizaciji i natjecanjima se može i mora popraviti."

Zadatci sa starijih natjecanja iz informatike trebaju biti korišteni pri poučavanju struktura podataka i algoritama, kako je i predlaženo u (Szuecs, 2001). Također, natjecanje predstavlja okruženje u kojem bi trebalo uvoditi i moderne aspekte programiranja, kao što su objektno orjentirani dizajni (Andrianoff et al., 2004).

Kako se može i primijetiti iz ranijeg opisa, natjecanja iz informatike u Hrvatskoj su doživjela znatan razvoj u posljednjih 15-ak godina. Među najnovija natjecanja pripada Hrvatsko otvoreno natjecanje u informatici, čija posebnost leži u tome što se odvija preko interneta, bez nužnosti boravka u unaprijed određenom prostoru. Takav oblik natjecanja je trenutno u svijetu vrlo moderan te je interes za takva natjecanja u velikom porastu, a više o tome zainteresirani čitatelj može naći u radu (Astrachan et al., 2003). Ovim Hrvatska u natjecanjima slijedi i moderne trendove.

Za kraj se kao idealan nameće zaključak iz rada (Dagienè, 2009, str. 1): „Iako informatika u mnogim zemljama nije poučavana kao zasebna disciplina, učenici su pozvani da sudjeluju u različitim natjecanjima iz informatike koja se organiziraju diljem svijeta. Kada se zainteresiraju za ova natjecanja, učenici žele dalje trenirati i dobiti odgovarajuću informatičku edukaciju. Time se natjecanja izdvajaju kao iznimno vrijedna za motivaciju i uključivanje učenika u nastavu informatike."