Informatics in Education, 2006, Vol. 5, No. 1, 125–132

Computer Science Contests for Secondary School Students: Approaches to Classification

Wolfgang POHL

Federal Contest in Computer Science
Ahrstr. 45, D-53175 Bonn, Germany
e-mail: pohl@bwinf.de

Received: March 2006

Abstract. The International Olympiad in Informatics currently provides a model which is imitated by the majority of contests for secondary school students in Informatics or Computer Science. However, the IOI model can be criticized, and alternative contest models exist. To support the discussion about contests in Computer Science, several dimensions for characterizing and classifying contests are suggested.

Key words: contest classification, Computer Science contests, science olympiads, Informatics olympiads

1. Introduction

International computer science (or informatics) contests for high school students traditionally concentrate on “algorithmic programming”, following the model of the International Olympiad in Informatics, see (IOI home page). I.e., students are required to solve algorithmic problems and implement the solutions as computer programs. For grading the students’ work, programs are checked for correctness by automatically testing their input/output behavior.

This approach has its benefits. In particular, the automatic grading process – in contrast to a manual review process – is less prone to subjectivity; mathematically taken, an evaluation function is implemented. However, there are also shortcomings. An automatic process may fail to be fair, too: Small mistakes may lead to complete failure, while other mistakes may not be discovered, and systematically so. Moreover, automatic systems may not be able to evaluate all possible aspects of a solution; e.g., quality of thought and clarity of expression are such problematic dimensions.

Therefore, it is worthwhile to consider both alternatives to and further development of the currently standard approach to Computer Science contests. In this volume, this will be done in several papers. In this brief paper, we will try to set the arena for such considerations. First, a short excursion is dedicated to identifying the differences between competitions, contests, and challenges. Second, we will sketch the landscape of computer science competitions for (high school) students. On the national level in particular, there is an interesting variety of contest approaches. In order to be able to talk about that variety, several dimensions for characterizing Computer Science contests will be suggested.

2. Competitions, Contests, Challenges – and Education

The reason for organizing competitions are diverse. On the surface, competitions provide means to identify excellence: A *competition* has one or more winners, who are assumed to be generally able to perform well in the area of the competition. Perhaps, sports championships provide the most exciting example of such competitions. But in sports, also other aspects of competitions show: People enjoy to compete, and even more people enjoy watching competitions and find them the more exciting the later the result of the match, the race, etc., becomes obvious. That is, a competition (as defined here) is organized for entertainment, too.

But there is more about competitions: They can also be contests or challenges, and such variations on the competition theme are more important for education. In a *contest*, and even more in a *challenge*, the main interest is in the quality of the individual performance, and the winners are determined by comparing absolute performance assessments instead of looking at the relative results of a directly competitive behavior. Contestants are confronted with problems, while competitors are confronted with each other.

In education, confronting students with problems is typically used to test the students' level of knowledge and expertise. In this case, the problems also demonstrate the expectation of the problem setter with respect to what level the students should have achieved. Thus, setting problems is a means of setting or expressing educational standards. Furthermore, students – gifted students in particular – can also be challenged by problems that cannot be solved by only applying learned mechanisms, but require special talent, mental abilities, and probably extraordinary effort, too.

While pure testing may be exciting enough for educators, it often raises more fear than excitement in the students. Therefore, contests go beyond educational testing by introducing a competition aspect into the test (hence the word con-test). By applying comparison, identifying winners, and awarding the students' achievements instead of just grading them, student motivation is increased, and fun and excitement return to the scene. In addition, when participation and achievement in a contest is published, contestants can reach a certain degree of fame. It is my personal experience with contestants that this popularity aspect very often adds to their motivation.

3. The Contest Space

For Germany, an overview of contests related to Computer Science (Pohl, 2005) organizes the mentioned competitions into the following groups: project contests, task contests, robot contests, and others. This grouping is based on the national reality of existing contests, but does not provide a sound classification. We will now describe some dimensions that might be more appropriate for classifying Computer Science contests.

3.1. *Scientific Area*

Although most Computer Science contests for secondary school students are programming contests with a focus on algorithmics, several other scientific areas can be identified that contests relate to. The perhaps most prominent examples are the diverse contests in Robotics; for secondary school students, examples are Robocup Junior (Robocup-Junior home page) and First Lego League (First Lego League home page). In Germany, the "Invent a Chip" (Pohl, 2005) contest deals with circuit design. Furthermore, there are mixed contests that cover different areas; an example is the annual contest of the American Computer Science League (ACSL home page). Contests with different subject divisions like the Imagine Cup (ImagineCup home page) can be seen as a collection of different contests. Some contests are completely open and allow the contestants to determine the subject of their work; an example is the European "Young Scientists Contest" (Young Scientists home page).

3.2. *Projects versus Tasks*

This is a very basic dimension: On the one hand, the organizers of a competition may invite contestants to demonstrate their own projects. In Germany, "Jugend forscht" (similar to the European Union "Young Scientists Contest", see (Jugend forscht home page)) is the classical example of a project contest. Contestants are free to choose the topic they would like to work on or the goals they want to achieve. Typically, they work at home on their projects, and at a "contest round" they are required to present their work at a presentation booth, which is visited by a jury and sometimes by the general public, too. Hence, besides the scientific quality of the work, presentation abilities may become a factor of success.

On the other hands, the majority of contests in general and computer science contests in particular, have the contestants work on given tasks. For instance, all the computer science contests mentioned in (Pohl, 2004) are task-based competitions. Typically, tasks are developed by task committees.

Sometimes, also intermediates between project and task competitions can be found. For instance, the "Javastars" competition (Javastars home page) calls for projects within a given framework: submissions have to be Java applications for learning, edutainment or infotainment, which can be utilized for school lessons. Such a competition model can be called "limited project contest".

3.3. *Short Time (Exam) versus Long Time (Homework) Rounds*

In many task-based competitions, contestants work on the given problems within a fairly short period of time. For instance, the programming exams of the IOI usually last 5 hours each. Within these five hours, three problems need to be solved. In project competitions, in contrast, contestants typically have ample time to work on their project and prepare the presentation for a contest round.

But there are also task-based contests with long time homework rounds, like the German "Bundeswettbewerb Informatik" (short: BWINF; Engl.: Federal Contest in Computer Science, see (BWINF home page)). BWINF contestants have about ten weeks time to deal with the first round tasks, and about 18 weeks time to get done with the tasks of the second round. Correspondingly, contestants are expected to deliver detailed written material, in most cases accompanied by a substantial piece of software.

3.4. *Grading: Automatic versus Manual*

Any contest needs to involve a grading procedure. Contributions to project-based contests need to be judged individually by a jury; we speak of pure manual grading in such a case.

In contrast, as described in (Pohl, 2004), most national and international computer science contests now involve completely automatic grading, performed by a computer system. Such a computer-based "grading system" nowadays typically receives the source code of a solution to a problem (which required the contestant to develop a short program). Then, it compiles this source code and executes it on a pre-defined set of test data. This model is applied at the most important contests, like IOI and the college-level contest ACM-ICPC.

However, these extremes are not without alternative. Task-based contests can incorporate grading schemes that prescribe how solutions to tasks should or even must be graded. In a BWINF (s.a.) contest, for each task a set of critical issues is defined. For each such issue, its weight and influence on the grading are prescribed. A contestant's work on a task is manually analyzed, according to such a grading scheme, by a number of university students of Computer Science. Such a grading process can be called schematic.

In addition, there are specialized contests that can apply even different grading procedures. For instance, the CodeCup contest (van der Vegt, 2006) requires contestants to write a program that can play a game against another program. The winner of the CodeCup is then determined by letting participants' programs play against each other. That is, the grading is done based on a comparison of performance. Therefore, we should better speak of a competition here, not of a contest. Other examples for "real" competitions are robotic soccer competitions like that of RoboCup (Junior).

3.5. *Submissions: From Hardware to Written Explanations*

Traditional Computer Science contests for secondary school students involve the creation of some piece of software. In recent years, however, many contests were founded that require the contestants to work with or even design and construct some hardware. The most prominent example perhaps is the Robocup contest with its Robocup Junior offspring. The Robocup Junior has three divisions: RoboSoccer – "teams" of one or two robots play football against each other, RoboRescue – robots perform a specific task within an emergency scenario, and RoboDance – groups of robots are involved in an "artistic performance". Several robot contests have been founded in recent years; (Pohl, 2005) mentions a number of robot contests that are held in Germany, Europe, and the world.

Another contest in Germany is “Invent A Chip”, where students are required to answer questions about Microelectronics, develop ideas for new chips and (the best are given that opportunity) do a software-aided chip design.

On that same dimension, i.e., the dimension of what contestants can submit, there are further possibilities. Project contests will typically not limit the format of the project itself; however, a description or presentation of the project is often required as the first submission. In typical programming competitions like IOI, other regional olympiads, and many national competitions as well, contestants answer the given tasks by submitting a program. In former years, executables were required, but recently, source code submission has taken over. This is not simply a change in procedure because, in principle, source code submission allows for non-automatic grading procedures. There are several competitions that mainly require the students to answer questions explicitly. One example is the Lithuanian “Beaver” contest, which is currently being transferred to other countries as well (Dagiene, 2006). To answer Beaver questions correctly, contestants sometimes need specific knowledge, but more often have to prove their understanding of informatics concepts as well as their ability of algorithmic and logical thinking. In the previous Beaver contests, there were many multiple-choice questions, but sometimes the contestants had to produce some result value. Within these limits, Beaver still allows for automatic grading. Another answer format is a written explanation; in homework task contests like BWINF, the written explanation of solution ideas is considered more important than the software (both source code and executable) that is submitted in addition. However, in general, written explanations cannot be graded automatically; for BWINF, a schematic manual grading process is used.

The contests of the American Computer Science League (ACSL) mostly consist of a “short answer test” and a programming problem, so that they do not take a clear position within this dimension. A short answer test contains five questions from categories like Number Systems, Boolean (Logic), LISP, Data Structures, Graph Theory, Digital Electronics, and WDTPD (What Does This Program Do). Typically, answers are really short and may often consist of a single number only. The programming problem is solved by submitting a program source code. Overall, an ACSL contest does not take a clear position according to this dimension. A programming problem may be solved within 72 hours, so that an ACSL contest as a whole cannot be classified clearly along the homework versus exam category either (ACSL home page).

3.6. *Age and Other Divisions*

Some competitions are divided according to the age of their contestants (e.g., the Robocup Junior contest, the ACSL contests, and the “Jugend forscht” project contest of Germany), while others – like the IOI – are not. This is perhaps not a dimension for contest classification, since one can view the introduction of age divisions as (virtual) introduction of a set of different contests (of the same kind) with different age limits. However, dividing a contest based on age is a feature that we consider worth mentioning. Actually, the age limit of a competition is perhaps one of its most crucial features. In task contests, problem setters will have to consider the age and assumed previous knowledge of contestants very seriously. Typically, for contests with only an upper age limit, it can be observed that tasks are designed to discover the most excellent students, whose age mostly is close to the limit; examples are IOI and BWINF. In a contest with upper and lower limit to the age of participants (which may be an age division of a larger contest), the tasks are much more tailored to fit the expected contestant population; the ACSL contests are an example for that.

Note that the USA Computing Olympiad (USACO home page) implements the concept of “difficulty divisions”. The contestants are not separated by age, but by their ability to solve the given problems. Based on their success, some contestants may be promoted to the division on the next difficulty level, and the main achievement in one year’s USACO is to be among the best contestants of the most difficult division. Hence, the lower divisions mainly serve to lower the threshold for entering into the whole competition and to practice for the most difficult division. BWINF has recently pursued a similar idea by giving a “junior problem” in the first contest round, which is easier than the other problems (but, however, may be solved by younger contestants only; i.e., age and difficulty get mixed here).

Furthermore, contests could be divided based on gender, like it is typically done in sports competitions. We do not know of an example for a Computer Science contest with gender divisions. Boersen and Phillipps (2006) describe the “Young Women’s Programming Contest” (YWPC) of New Zealand, which is exclusively dedicated to female participants.

4. Conclusion

Several dimensions for characterizing Computer Science contests were presented that lead to a classification scheme. They are summarized in Table 1. Using these dimensions, the International Olympiads in Informatics can be classified as task competitions with short time exam setting and automatic grading of software submissions. However, there

Table 1

Dimensions for characterizing computer science contests

Dimension	**Values**
Scientific Area	Algorithmics, Robotics, Chip Design, …, Mixed, Open
Style	Project, Limited Project, Task, Mixed
Duration	Short Time (Exam), Long Time (Homework), Mixed
Grading	Automatic, Schematic, Manual
Submission	Hardware, Programmed Hardware, Executable Program, Program Source Code, Answer Value, Explanation, Description/Presentation, Mixed
Divisions	Age, Difficulty, Gender

are many alternative contest models, and many examples of different contest types were presented. In a discussion about developing and improving IOI and other contests in Computer Science, the suggested dimensions will hopefully provide a helpful framework and guidance as well.

As another characteristical dimension for contests in general, also the form of organization could be discussed. In Computer Science in particular, many contests are organized as Internet contests (like USACO, with the exception of the final round) or offer an Internet contest in parallel to an on-site contest with personal presence (like IOI). Other possibilities are to have the contestants work individually at home (like BWINF) or in a coordinated way, e.g. at school (like Beaver), where solutions may be submitted to some central authority or some local organizer. We do not regard the form of organization as a characteristical dimension. However, the way a contest is organized typically restricts the possible values of the dimensions we defined above. For instance, Internet contests typically require electronic submission channels, so that hardware submissions are not possible. Furthermore, contests with automatic grading are much easier to organize as Internet contest than others.

References

ACSL home page.
`http://www.acsl.org/`.

BWINF home page.
`http://www.bwinf.de/`.

First Lego League home page.
`http://www.firstlegoleague.org/`.

Imagine Cup home page.
`http://imagine.thespoke.net/`.

IOI home page.
`http://www.ioinformatics.org/`.

Javastars home page.
`http://www.javastars.de/`.

Jugend forscht home page.
`http://www.jugend-forscht.de/`.

Robocup Junior home page.
`http://www.robocupjunior.org/`.

USACO home page.
`http://www.usaco.org/`.

Young Scientists contest home page.
`http://europa.eu.int/comm/research/youngscientists/`.

Boersen, R., and M. Phillipps (2006). Programming contests: Two innovative models from New Zealand. In W. Pohl (Ed.), *Perspectives on Computer Science Competitions for (High School) Students*.
`http://www.bwinf.de/competition-workshop/papers.html`.

Dagiene, V. (2006). Competition in information technology – learning in an attractive way. In W. Pohl (Ed.), *Perspectives on Computer Science Competitions for (High School) Students*.
`http://www.bwinf.de/competition-workshop/papers.html`.

Pohl, W. (2004). National computer science contests.
`http://www.bwinf.de/olympiade/national-contests.pdf`.

Pohl, W. (2005). Informatik-Wettbewerbe in Deutschland. *LOG IN*, **133**, 10–23.

van der Vegt, W. (2006). The CodeCup, an annual game programming competition. In W. Pohl (Ed.), *Perspectives on Computer Science Competitions for (High School) Students*.
`http://www.bwinf.de/competition-workshop/papers.html`.

W. Pohl was educated in computer science, and received a PhD in 1997 from the University of Essen, Germany. For many years, he investigated the use of artificial intelligence techniques for the improval of interaction between humans and machines. In 1999, he changed position and perspective by becoming executive director of the German Federal Contest in Computer Science. Among his responsibilities is to coach the Germann IOI team and lead the German IOI delegation. Now, his interest lies in improving computer science contests, establishing new ones, and working on diverse other projects, everything in order to popularize computer science among youth. In 2003, he was elected member of the IOI International Committee for three years, and currently holds the position of executive director of IOI.

Vidurinės mokyklos mokinių informatikos varžybos: galima klasifikacija

Wolfgang POHL

Šiuo metu didžioji dalis vidurinėse mokyklose organizuojamų informatikos varžybų imituoja Tarptautines informatikos olimpiadas. Vis dėlto šis modelis gali būti kritikuotinas – esama gana gerų alternatyvių modelių. Siekiant paskatinti aplink moksleivių informatikos varžybas besirutuliojančias diskusijas, pateikiama keletas galimų papildomų varžybų klasifikavimo bei charakterizavimo aspektų.