The IOI Is (Not) a Science Olympiad

Tom Verhoeff*

October 2005

—

Keywords: informatics competitions, programming contests, computer science

Abstract

The International Olympiad in Informatics (IOI) aspires to be science olympiad alongside the international olympiads in mathematics, physics, chemistry, and biology. We argue that, in its current format, the IOI fails to be scientific in an important way. Informatics as a discipline is well suited to a scientific approach and it offers numerous possibilities for competitions with a high scientific standing.

In this paper, we describe the major ingredients of the IOI to guide further discussions. By presenting the results of an extensive analysis of two IOI competition tasks, we hope to create an awareness of the urgency to address the shortcomings. We offer some suggestions to raise the scientific level of the IOI.

1 Introduction

The International Olympiad in Informatics, abbreviated IOI, is an annual competition in the discipline of computer science (IOI (2005)). It was established in 1989 and modeled after the International Mathematical Olympiad. The IOI is aimed at pupils in secondary education, especially those talented in computer science. One of the main objectives of the IOI is "to bring the discipline of informatics to the attention of young people" (Statute S1.7 from the IOI Regulations).

The IOI competition has evolved over the years to its current format of a programming contest. We will show that this has led to a number of undesirable features, which conflict with the aspiration of being a member of the family of science olympiads and which interfere with its main objectives. As a consequence, the IOI may fail to attract the talented students that the discipline of informatics needs and it fails to promote the science of computing in the best way possible.

---

*Technische Universiteit Eindhoven, The Netherlands, `T.Verhoeff@TUE.NL`

1.1 Overview

In Section 2, we describe the IOI competition in more detail, in particular how its informatics content has evolved. Section 3 concerns an extensive analysis of two IOI competition tasks and a re-evalution of the submitted programs. In Section 4, we summarize the main difficulties in grading algorithms. Section 5 offers some ideas to improve future IOI competitions.

2 The IOI Competition

The current *IOI Regulations* (IOI (2005)) define the IOI as *an annual international informatics competition*. However, these regulations do not otherwise constrain the actual content of the competition. They do constrain the form and organization.

Each year, the Host of the IOI is obliged to produce *Competition Rules and Judging Procedures*. These rules and procedures offer some further insight into the content of the competition. They have evolved over the years, though there is an intent to strive for continuity. Further background information about the IOI competition can be gleaned from ISC (2000).

The regulations of the first IOI (IOI, 1989, p.5) are more explicit than the current regulations. We quote:

1.3 "The IOI consists in solving one problem by using personal computer (PC) and it is carried out in one day within four hours.

1.4 "The problem will be selected by the International Jury (see 2.1) on the basis of the problems, provided in advance by the participating countries.

1.5 "The problem will be independent of the PC hardware and high level programming languages can be used for its solution (e.g. BASIC, LOGO, PASCAL, etc.). The problem will be of *algorithmic nature* [emphasis added] and no specialized software packages etc. will be necessary to solve it.

1.6 "The complete solution of each problem includes

   (a) An informal and block-diagram description and *argumentation* [emphasis added] of the algorithm on the basis of which the program is written;

   (b) The source text and results of the program execution, obtained by the microcomputer (2 copies);

   (c) The floppy disk on which the final version of the program is stored, in the source language and eventually in executable code (2 copies)."

The Competition Rules for IOI 2005 state

"All of the tasks in IOI 2005 are designed to be algorithmic in nature. Efficiency plays an important role in some tasks. Whenever efficiency

> of an algorithm is important, certain correct but inefficient approaches can score points ..."

"Algorithmic in nature" is still rather vague. Current practice is that (most of the) competition tasks present problems which have an algorithm as solution. The algorithm has to be expressed in one of the approved programming languages (currently, Pascal, C, and C++). The historic motivation for requiring a machine-executable implementation is along the following lines:

- There are no standardized, well-known, abstract algorithm notations[1] (let alone notations within reach of the IOI competitors).

- IOI competitors can be expected to be familiar with at least one popular programming language.

- The syntax and semantics of standardized programming languages are well defined.

- Some qualities of machine-executable programs can be tested automatically.

Note that point 1.6(a) in the IOI'89 Regulations was soon dropped. The main reason being that competitors had little means to express their arguments, and grading of these arguments turned out to be labor intensive.[2] Remember that in those days informatics was not a regular topic in secondary education of most countries. Competitors were mainly self-taught.

Since IOI'94, grading of the programs submitted by the competitors has been automated. Initially, competitors created the executables themselves and no longer submitted source files (disadvantages: competitors had to know about compiler options etc.; no source files available for further analysis; advantage: competitors were in full control of compilation). Since IOI'01, competitors are required to submit source files, and the organizers handle compilation.

The grading process nowadays is roughly as follows:

1. If the submitted source file does not compile, then the score is zero; otherwise, the resulting executable is evaluated further.

2. The executable is run on each of a number of inputs, while its execution is monitored and the output is captured.

3. If an execution raises an exception (run-time error; e.g. index out of bounds, division by zero, time-limit exceeded), then that particular run scores zero.

---

[1]This can be viewed as a weak spot of informatics.

[2]Compare this to the grading at the International Mathematics Olympiad (Verhoeff (2002)), where some 60 mathematicians grade the work on six problems by about 500 competitors over a period of two days.

4. If an execution terminates normally, without violating any of the resource constraints, then the output is checked for correctness, resulting in a score for the run.

5. The final score is obtained as the sum of the scores for the individual cases. At IOI'05, test runs were clustered in test cases, where the score for a test case is the *minimum* of the scores for the constituting test runs.

The organizers determine the test cases (their number, contents, and maximum score) before the competition. These test cases are not known to the competitors during the competition. The organizers also set resource limits in advance, in particular, on the total execution time and the total data memory. Other resource limitations typically are: single thread (no "forking"), single processor, no auxiliary files, no network access. It also turned out to be necessary to limit the compilation time.[3]

The number of test cases and their design have evolved somewhat. At IOI'94, the number ranged from four to eight test runs per task. At IOI'05, it ranged from 20 to 24 cases, with many cases consisting of two runs. Since IOI'04, the design of the test inputs is specifically split into *testing for correctness* with "smaller" inputs and *testing for efficiency* with a variety of more demanding inputs to distinguish relevant efficiency classes (ISC (2004)).

2.1 Summary of Constraints on the IOI Competition

As an aid to understand how much freedom there is in setting up an IOI competition, we summarize various constraints. Some of them are non-negotiable, whereas others do leave some room for interpretation. We list in no particular order:

1. The subject matter is informatics, also known as computing science or computer science; in particular, it should include the areas of algorithms and data structures.

2. The competitors are in secondary education from all over the world.[4]

3. The competition tasks should focus on problem solving (and not, e.g., rote learning of encyclopedic knowledge).

4. The problem statements should be elementary to understand, i.e., require no specific prior knowledge, other than what is common at the level of secondary education. Also see Verhoeff (2004).

5. It should be possible to obtain solutions by elementary means, i.e., this should not require special skills (e.g. speed typing) or specific prior knowledge (e.g. computer architecture).

---

[3]Through the use of macros, some of the computations can be diverted to the compiler. Furthermore, compilation is done on a server, shared by all competitors.

[4]The IOI Regulations provide a more precise definition.

6. Work submitted by the competitors should be effectively gradable within the schedule and budget of the IOI.

7. The competition tasks should contain some innovative element.

8. The competition tasks should provide diversity in level of difficulty, problem domain, and applicable solution techniques.

Some further points to keep in mind are:

- The IOI competition is carried out like an exam, where the competitors work individually on some tasks and at the end submit their work for evaluation, resulting in a score by which the individuals are ranked linearly.

- The IOI competition has a responsibility to promote the discipline of informatics (in a positive way); it is not intended to be recreational or for entertainment only.

- There is a natural language barrier to be crossed twice: both from organizers to competitors and from competitors to organizers. We cannot rely on a common natural language.

3   Some Results of Analyzing Past IOIs

As a master's thesis project, Wouter van Leeuwen analyzed two IOI competition tasks, together with the submitted work for these tasks and the scores obtained at the IOI (v.Leeuwen (2005)). His research concerned the tasks Median (IOI 2000) and Phidias (IOI 2004):

Median Find the object of median strength among an odd number of objects with distinct strengths, using only the function *Med3* that returns the object of median strength among *three* distinct objects. It is cast as a reactive task, where the program to be designed communicates with an environment implemented as a library.

Phidias Determine the minimum wasted area, when cutting a marble slab to obtain plates from a given set of desired dimensions. The cuts are done sequentially and they run either horizontally or vertically, going all they way from one side to the opposite side.

The details of these tasks does not matter much for the discussion here.

3.1   Analysis of Results for Median

Table 1 shows how many competitors (of the 209 that did submit a source file)[5] obtained what score. Since there were ten test inputs, each with a maximum score

---

[5]Of the 274 competitors, 226 submitted something. These numbers include the second team from the Chinese host.

of ten points, the scores range from 0 to 100 in steps of 10. The right-hand column gives the total number of points obtained by these competitors together. The task Median was a harder task with only 10% of the competitors obtaining a score of 90% or more.

Score	100	90	80	70	60	50	40	30	20	10	0	Total
IOI 2000	24	6	3	3	16	5	11	16	15	59	51	6410
v. Leeuwen	18	6	1	0	15	2	8	14	5	12	128	4380

Table 1: Frequency distribution of scores for task Median (IOI 2000)

A detailed analysis of the task Median and the design of the test inputs used at IOI'00 is given by Horváth, Verhoeff (2002). This task appears to have a rich solution space. The analysis of van Leeuwen, however, showed that the competitors were able to find even more interesting algorithms. This was not discovered during IOI'00, because of the automatic grading, which treats the submissions as black boxes. We will give an example in Section 5.

The re-evaluation done by van Leeuwen was aimed primarily at correctness. Note that there are two ways in which competitors lose points: becuase of incrrectness and because of inefficiency. The difference cannot be seen in the statistics. It turned out that under IOI grading, 99 competitors submitted programs that produced an incorrect output for at least one of the test cases. Furthermore, 67 of these had a non-zero score. From the IOI grading is completely unclear what kind of correctness errors were made, and how they should be ranked with respect to each other. The research by van Leeuwen uncovered another 10 incorrect submissions, that were not identified as such by the IOI grading. Note that van Leeuwen decided to be quite strict in his re-evaluation, where he assigned a zero score to all submissions that were incorrect.

From Table 1, we see that over 30% of the total score has not been properly justified. Why would one incorrect program obtain a higher score than another. The test cases hardly say a thing about the kind of correctness error. Even for efficiency, there are scores that one can doubt, because the 18 remaining solutions that score 100% are not sufficiently efficient in the worst case, that is, they do not solve all allowed inputs within the limits.

Because of its reactive nature, the task Median allows for some additional automatic evaluation techniques:

- By analyzing the log file that records the dialog, it is possible to do stricter correctness testing (e.g., detect bluffing), and algorithms can be classified through their behavior *during* a test run.

- By using input values that depend on previous outputs, it is possible to provoke (near) worst-case behavior (a so-called adversary).

- Resource limitations can be enforced more objectively.

6

3.2 Analysis of Results for Phidias

Table 2 shows how many of the 295 competitors[6]) obtained what score. Since there were 20 test inputs, each with a maximum score of five points, the scores range from 0 to 100 in steps of 5. The right-hand column gives the total number of points obtained by these competitors together. The task Phidias was a medium task with 35% of the competitors obtaining a score of 90% or more. By that standard, it was the easiest task at IOI'04.

Score	100	95	90	85	80	75	70	65	60	55	50
IOI 2004	76	12	6	9	1	5	12	6	15	9	19
v. Leeuwen	48	10	0	8	0	3	5	3	8	3	0

Score	45	40	35	30	25	20	15	10	5	0	**Total**
IOI 2004	4	8	7	22	5	2	3	7	7	60	**15795**
v. Leeuwen	0	0	0	0	0	0	0	0	0	207	**7845**

Table 2: Frequency distribution of scores for task Phidias (IOI 2004)

Under IOI grading, 104 competitors submitted programs that produced an incorrect output for at least one of the test cases. Furthermore, 89 of these had a non-zero score. The research by van Leeuwen uncovered another 57 incorrect submissions, that were not identified as such by the IOI grading. From Table 2, we see that over 50% of the total score has not been properly justified.

4 The Difficulties of Grading Algorithms

Section to be completed after recovering from illness.
 Edsger Dijkstra (Dijkstra (1972)):

> "Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence.
>
> The only effective way to raise the confidence level of a program significantly is to give a convincing proof of its correctness."

In fact, a program text by itself is quite uninteresting. The reasoning behind the design is what really matters, especially from a scientific point of view. The same holds for showing (upper bounds on) the worst-case, average-case, or best-case performance.

5 Ideas for Future Directions

Even when we stick to competition tasks that are "algorithmic in nature", there are numerous alternatives to a programming contest. For instance, a task could involve

---

[6]These numbers include the second team from the Greek host.

one or more algorithms, but not as its solution.

Here is a new algorithm for solving task Median of IOI'00, which van Leeuwen discovered among the submissions. We named it *Bubble Search* because of its resemblence to Bubble Sort.

1. Put all N objects in a sequence, say $a[i]$ for $i \in \{1, \ldots, N\}$

2. Let $M = (N + 1)$ **div** 2

3. Repeat the following for-loop, until no swap has occured in a

   > For each $i \in \{1, \ldots, M - 1\}$ do
   > i. Let $x = Med3(a[i], a[M], a[N - i + 1])$
   > ii. If $x \neq a[M]$ then swap x and $a[M]$ in a

It is obviously correct, *if* it terminates, because $a[M]$ has not moved and equally many objects surround it. But it is far from obvious why the algorithm terminates. Without termination argument, the algorithm is merely a bluff. It is a nice algorithmic problem to discover why it terminates. We posed this problem to the candidates for the German delegation at IOI'05. Somewhat to our surprise, they were very capable to formulate (either in German or English) proper arguments for terminations. Other questions they addressed were: what is the worst-case performance (number of *Med3* calls as function of N); what is a witness input for a worst-case; what is the average-case performance. The former two problems were also solved. The latter problem was not really solved, though they did carry out some programmed experiments and did some curve fitting.

To be expanded...

6 Conclusion

We have shown that the current grading practice of the IOI has severe shortcomings. It is a (well-known) mistake to evaluate the correctness and efficiency of an algorithm solely on the basis of executing the implementation on a set of test cases. It is reasonable to doubt the validity of the results of the IOI from a scientific point of view. Analysis of two IOI tasks and the submitted work confirms these doubts.

The computer science goals of the IOI should be stated more explicitly. In particular, the meaning and value of correctness and efficiency should be better established. Is there a reasonable way to rank two incorrect algorithms? What about ranking algorithms by worst-case, average-case, and best-case efficiency?

Within the area of algorithmics, there are various other types of competition tasks that could be posed. Grading of the submitted work may not always be done automatically. But the desire for automation as carried out in the past has blinded the IOI community: the real merits of the work of the competitors remained invisible. Even in the case of "traditional" IOI tasks, it would be advisable to base the result not only on the opinion of a preprogrammed automaton.

References

Dijkstra, E. W. "The Humble Programmer", *Communications of the ACM*, **15**(10):859–866 (1972).

Horváth, G. and Verhoeff, T. "Finding the Median under IOI Conditions", *Informatics in Education*, **1**(1):73-92 (2002).

Kenderov, P. S. and Maneva, M. N. (Eds.) *Proceedings of the International Olympiad in Informatics*, Pravetz, Bulgaria, May 16–19, 1989. Sofia: Union of the Mathematicians in Bulgaria, 1989.

IOI, *International Olympiad in Informatics*, Internet WWW-site, URL:
`http://www.IOInformatics.org/` (accessed October 2005).

ISC (IOI Scientific Committee). *Guidelines for IOI Competitions*. Version 0.3, Jan./Feb. 2000.
`http://olympiads.win.tue.nl/ioi/sc/documents/ioi-sc-guide-03-nr.txt`

ISC (IOI Scientific Committee). *Principles behind IOI 2004 Competition Tasks and Their Grading*.
`http://olympiads.win.tue.nl/ioi/sc/documents/principles-2004.txt`

van Leeuwen, W. T. *A Critical Analysis of the IOI Grading Process with an Application of Algorithm Taxonomies*, Master's Thesis, Technische Universiteit Eindhoven, Faculty of Mathematics and Computing Science, October 2005.
`http://www.win.tue.nl/˜wstomv/misc/ioi-analysis/thesis-final.pdf`

Verhoeff, T. *The 43rd International Mathematical Olympiad: A Reflective Report on IMO 2002*, Computing Science Report 02-11, Faculty of Mathematics and Computing Science, Technische Universiteit Eindhoven. August 2002.
`http://www.win.tue.nl/˜wstomv/publications/imo2002report.pdf`

Verhoeff, T. *Concepts, Terminology, and Notations for IOI Competition Tasks*, document presented at IOI 2004 in Athens, 12 Sep. 2004.
`http://olympiads.win.tue.nl/ioi/sc/documents/terminology.pdf`