Olympiads in Informatics, 2009, Vol. 3, 144–148



# Using Subtasks

Willem van der VEGT
*Dutch Olympiad in Informatics*
*Windesheim University for Applied Sciences*
*PO Box 10090, 8000 GB Zwolle, The Netherlands*
*e-mail: w.van.der.vegt@windesheim.nl*

**Abstract.** In the Dutch Olympiad in Informatics subtasks are used to create a nice score distribution and to reward contestants for what they were able to solve. Using subtasks can provide a mechanism to distinguish between contestants at the International Olympiad in Informatics.

**Key words:** informatics olympiad, programming competition, task design.

## 1. Introduction

The Dutch Olympiad in Informatics (Dutch, 2009) takes three rounds. The first round tasks are published on our website; contestants can work on these tasks for several months, they are allowed to cooperate and use a language of their choice. The annual CodeCup (CodeCup, 2009) task is one of the tasks of this first round. Contestants with a result that is somewhat better then solving just one task are invited for a one day contest at a university. This second round is very selective; we want to identify the top ten students, but we also intend to offer a fair competition in which contestants can show what they are able to do in a few hours.

The top ten students get a trainings course for three days and usually they enter the USACO (USACO, 2009) to get more practice experience. The selection phase ends with a one day competition with three or four IOI-style tasks.

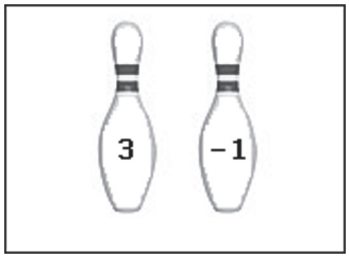
This paper addresses the way we organize our second round tasks. We make intensive use of subtasks, to get a nice score distribution and to allow all contestants to score at least some of the points. This has a lot of advantages, but of course there are also some flaws. We will discuss as examples the tasks "Bowling with numbers" and "Maximum height".

In the past IOI-competitions subtasks have been an item. In Athens (Athens, 1991) and Bonn (IOI'92 Problems, 1992) only one task was presented on each competition day. These tasks were manually graded and grades were given for different parts of the solution. In Eindhoven (The problems, 1995) and Vesprem (The IOI'96 Competition, 1996) grading was already automated; some of the three tasks of a competition day were split up in two subtasks. The subtasks have since disappeared; the introduction of the 50% rule was another way to distinguish between contestants. We think it is possible to use subtasks to realize the goals that are now served using different limits on test cases.

## 2. Task "Bowling with Numbers"

This was originally a task from the Canadian Computing Competition in 2007 (Canadian, 2007).

In short: A number of pins are placed in a row with equal distances. On each pin there is a number indicating the value of this pin. You get a bowling ball with a certain width and you are able to throw a few rounds. Maximize the total value of the pins you can drop. In the task description two players are mentioned: Alice plays perfect, Bob plays using a specified greedy strategy.




In Canada this task was used in two different sessions. In stage 1 of the competition it was a task with only positive numbers. But in stage 2 the problem was complicated by adding also negative values for the pins. In the Canadian classification system for olympiad tasks this second task was rated at the highest level.

In 2008 we used this task at the Dutch Olympiad in Informatics. We divided it into the following subtasks:

| | |
|---|---|
| Subtask A: | Most valuable pin. Determine the maximum value available on all of the pins. |
| Subtask B: | Theoretical maximum. Find the maximum score you can earn if you were able to throw an infinitive number of times with a ball of width one. |
| Subtask C: | Implement the specified greedy solution |
| Subtask D: | Find the optimal solution. |

None of the 19 contestants was able to find a program that produced an optimal solution for all test cases.

The score distribution was:

Table 1

Results for task "Bowling with numbers"

| Subtask | Maximum score | Average score | #contestants with maximum score | #contestants with partial score |
|---|---|---|---|---|
| A | 12 | 10.4 | 16 | 1 |
| B | 18 | 12.0 | 11 | 3 |
| C | 28 | 8.7 | 5 | 3 |
| D | 42 | 4.4 | 0 | 7 |
| Total | 100 | 35.5 | 0 | 17 |

## 3. Task "Maximum height"


In short: You get a graph with numbered places and connections. Every connection has a maximum heigth. You run a car service on this graph and you need to determine the maximum heigth of the vehicles that you can use, given certain constraints.

This task was submitted for IOI 2004. The task proposal had two subtasks. Since it was not used at IOI 2004, we used it in our national olympiad in 2005. This time it was split up in three different subtasks.

Subtask A: Cab service. What is the maximum height if you want to be able to use every road in the graph, i.e. find the minimal maximum height of all roads.

Subtask B: City service. For every city this service will bring you to all adjacent cities. If there is more then one immediate connection between two cities, you can ignore all but the one with the maximum height.

Subtask C: Overall service. All cities can be reached, but not necessarily by a short path. Now you will be looking for a spanning tree in which the minimal height of all connections is maximal.

Four contestants out of 42 were able to solve all subtasks.
The score distribution was:

Table 2
Results for task "Maximum height"

| Subtask | Maximum score | Average score | #contestants with maximum score | #contestants with partial score |
|---|---|---|---|---|
| A | 15 | 12.2 | 32 | 6 |
| B | 25 | 16.0 | 13 | 22 |
| C | 60 | 13.6 | 4 | 11 |
| Total | 100 | 41.8 | 2 | 37 |

## 4. Results and Other Experiences

In these two cases the use of subtasks worked rather well. We have got a nice score distribution, almost all contestants scored at least some points on these specific tasks. There is a large gap between the algorithmic complexity of finding a maximum number (subtasks A in both cases) and the optimal solution of an IOI-like task (as in the final subtasks).

There is a strong correlation between the order of the subtasks and the results of the participants. In these examples this has to do with the increasing difficulty of the subtasks. In an earlier olympiad however we used another task where four or five different algorithms were introduced to find a path in a given graph. On this occasion, one of the first subtasks was much more difficult then the other subtasks. Alas, almost no one tried to solve the easy subtasks, because the difficult one frightened them too much.

The system of subtasks also makes it easier to slip in a more theoretical subtask. Once we asked contestants to create a sample input file that could produce a specified output for a given algorithm. We only asked for this file, not for a program or a description how to find it. At the IOI we call this an output only task; we had just an output only subtask.

In this year's second round a game is played in which you can get stuck.

One of the subtasks is to output the minimal number of moves that is at least possible, whatever the initial position of the game, and however badly the game might be played. We even predicted the output for a specific case and asked the contestants to explain this strange output in a few words. In this case, they had to deliver a plain text file. Of course this file was examined and graded manually.

Sometimes subtasks are dependant. It is for instance not possible to solve subtask C without solving subtask B. In these cases it is fair to give some of the credits to those participants that are able to solve subtask B; they already found a part of the overall solution, but failed in the next step.

## 5. Subtasks at IOI

On several occasions we have spoken on new task types and competition ideas regarding the IOI. The idea of subtasks was not mentioned in the IOI-workshop 2006 (Report, 2006) though spreading the difficulty level was one of the topics discussed.

Using subtasks is one of the ways we can try to spread the difficulty level. The same was aimed by the former 50%-rule. This was intended to distinguish between correct programs and correct and efficient programs. The use of subtasks gives credit to those participants that solved a part of the problem. It also gives the possibility of incorporating small theoretical questions, like specific border cases, sample input or output. It will spread out the results, give more credit to the weak contestants but will still allow us to identify the very best and brightest.

## References

*Canadian Computing Competition* (2007).
`http://cemc.math.uwaterloo.ca/contests/computing/2007/index.html`
*CodeCup* (2009). `http://www.codecup.nl`
*Dutch Olympiade in Informatics* (2009). `http://www.informaticaolympiade.nl`
*Athens 3rd International Olympiad in Informatics* (1991). Athens, Greece, May 19–25.
`http://ioinformatics.org/locations/ioi91/tasks91.txt`
*IOI'92 Problems* (1992). Bonn, Germany, July. IOI 1992 Bonn.
`http://ioinformatics.org/locations/ioi92/tasks92.txt`

*The problems* (1995). IOI 1995 Eindhoven.
http://ioinformatics.org/locations/ioi95/contest/index.shtml
*The IOI'96 Competition* (1996). Vesprem.
http://ioinformatics.org/locations/ioi96/contest/index.shtml
*Report of the Competition Workshop* (2006). Dagstuhl.
https://phreax.net/bwinf-competition-workshop/wiki/index.php/Main_Page
*USACO* (2009).
http://contest.usaco.org/ioigate

**W. van der Vegt** is teacher's trainer in mathematics and computer science at Windesheim University for Applied Sciences in Zwolle, the Netherlands. He is one of the organizers of the Dutch Olympiad in Informatics and he joined the International Olympiad in Informatics since 1992. He was involved in the IOI-workshops on tasks in Dagstuhl (2006) and Enschede (2008). Currently he is deputy team leader of the Netherlands.