Some Thoughts on the Role of Competitions in Computing Science Education with a Twist in the Direction of Theoretical Computer Science

Tom Verhoeff
Dept. of Mathematics & Computing Science
Eindhoven University of Technology
P.O. Box 513
NL-5600 MB Eindhoven
Tom.Verhoeff@acm.org
http://www.win.tue.nl/~wstomv/

Computer Science is a relatively young discipline that has not (yet) established a firm position in the secondary school curriculum among other (older) disciplines (a situtation that holds in almost all countries). However, we go through a period of rapid change, to catch up, so to say.

One measure of acceptance and integration of a discipline into the curriculum is the situation regarding competitions in that discipline. A healthy, diverse set of competition events indicates that the discipline is well accepted. A lack of (good) competitions is a sign of poor acceptance.

The exact sciences have prestigious international olympiads for talented pupils from secondary schools. Here is a brief list:

IMO :	International Mathematics Olympiad (since 1959)
IPhO:	International Physics Olympiad (since 1967)
IChO:	International Chemistry Olympiad (since 1966?)
IOI:	International Olympiad in Informatics (since 1989)
IBO:	International Biology Olympiad (since 1990)
IAO:	International Astronomy Olympiad (since 1996)

I have been involved with the IOI since 1994. On one hand, the existence and healthy state (nearly 60 participating countries makes it the second largest after IMO) of the IOI is encouraging. On the other hand, the contents and operation sometimes makes me wonder why we cannot do better.

There are other prestigious contests in these disciplines as well, at various educational levels. For example, the "Putnam" is a well-known math competition in North America for college students,"First Step to Nobel Prize in Physics" is another physics contest for high school students, and the ACM International Scholastic Programming Contest (ICPC) has become a prestigious computing science competition for university students (with high-visibility sponsors that provide generous funding). By the way, the Finals of the ACM ICPC are planned to be held outside of North America for the first time during 1999 in Eindhoven, The Netherlands.

Getting back to CS again, the CS contests that I am aware of are all centered around programming. Sometimes the algorithmic nature is emphasized, but even then the contest usually involves actual implementation of algorithms as well. At the IOI, the main reason appears to be that for the participants (aged under 20, with little international communication skill), there is a language barrier. This barrier somehow needs to be crossed twice: once when posing the competition tasks and the second time when juding the results submitted by the participants. The second occurrence of the barrier can be circumvented by requiring the participants to submit their (algorithmic) solutions in computer readable format (i.e. as a computer program expressed in one of a few standardized programming languages). In fact, the executability of these solutions is exploited when judging the results: the programs are applied to a set of test cases and their output is judged.

To give you an idea of the kind of problems that are posed at the IOI, consider the first problem posed at IOI'94: *The Triangle* (with numbers) [http://www.win.tue.nl/ioi/ioi94/contest/day1prb1/]. (A nice feature of this problem is that its description is fairly short; it is not so difficult). This problem shows that just hacking together a program is not good enough; you need to think about the performance of your program. This is a more advanced skill than knowing implementation, machine, and OS details.

Some advocates of programming contests claim that good programming problem will require the participants to address all kinds of theoretical issues as well. Nevertheless, it is my feeling that we should be able to offer more variety. At IOI'95 we made an attempt, which was only partly successful. The final problem on the second contest day had a new twist to it. Traditionally, IOI programming problems have a minimal I/O interface, because the I/O complications are least interesting (from an algorithmic point of view). Before IOI'95 that had always meant batch processing: all input is available in an input disk file (organized in a simple way) and all output goes to an output disk file (with a similar simple structure). At IOI'95 we introduced an "interactive" I/O interface: *Wires and Switches* [http://www.win.tue.nl/ioi95/task/wires.html]. Here, the input received by the participant's program from the environment may depend on earlier output of the program itself. This problem brought up discussions (what if the environment is non-deterministic during the judging), but was generally well received.

Another experiment at IOI'95 concerned a "theoretical" problem without computer. This went less well, for one thing, because coaches knew how to prepare the participants for programming problems (due to availability of prior material), but not for theoretical problems. Another reason may have been the particular problem that we used. The reserve "theoretical" problem (for which we anticipated some other troubles): *Faulty Links* [http://www.win.tue.nl/ioi95/task/links.html], I still consider a nice problem (though a bit long, with two pages of reading material and another two pages of forms to fill out for answering it).

I can imagine some of the difficulties when trying to put together a TCS contest, but I am *not* convinced that it *cannot* be done. Little (certainly, too little) effort has been made in this direction. I would appreciate receiving any suggestions for theoretical problems that could be posed at an olympiad level (i.e. with little formal background knowledge) or even at a university level. It is quite a challenge to organize a good competition. Even chosing the final wording of contest problems is a delicate matter, which you may only appreciate after you have given it a try and then presented your description to a committee of critical (and sometimes perversely suspicious) coaches of IOI teams.

The lack of a good, interesting, and thriving TCS competition is an ominous signal that cannot be ignored. Either we do something about it and prove that it can be done (and, thus, do the field a badly needed service), *or* we must conclude that TCS is not suitable for this, and thereby also admit that there may be a ground for the students' general dislike of TCS. Fortunately, there are still those students that after a programming contest hate machine details enough to prefer a more abstract field.

It is my belief that the IOI should strive for more diversity, including non-programming CS problems. In order to make that happen, support from the (T)CS community is badly needed.