Olympiads in Informatics, 2008, Vol. 2, 105–114

Challenges in Running a Computer Olympiad in South Africa

Bruce MERRY
ARM Ltd
110 Fulbourn Road, Cambridge, CB1 9NJ, England
e-mail: bmerry@gmail.com

Marco GALLOTTA, Carl HULTQUIST
Department of Computer Science, University of Cape Town
Private Bag, Rondebosch 7701, Cape Town, South Africa
e-mail: marco@gallotta.co.za, chultquist@gmail.com

Abstract. Information and Communication Technology (ICT) in South Africa lags behind that of the developed world, which poses challenges in running the South African Computer Olympiad (SACO). We present the three-round structure of the SACO as it is run today, focusing on the challenges it faces and the choices we have made to overcome them. We also include a few statistics and some historical background to the Olympiad, and sample questions may be found in the appendix.

Key words: digital divide, olympiad in informatics, problem solving, South Africa, SACO.

1. Introduction

For historical reasons, the level of ICT infrastructure in South African schools spans a wide range. At schools in affluent suburbs, computers are available to students, Internet access is common, and most students can take optional classes in computer studies (either at their own school or a nearby centre). On the other hand, poorer schools lack the most basic of facilities, and students have no access to computers or the Internet. This is sometimes referred to as the *digital divide*.

This makes organising a representative computer olympiad challenging. We would like to involve as many students as possible, to foster interest in computer science and computer programming amongst talented students. But how can one run a computer olympiad for students with no access to computers? South Africa is larger than most European countries, so gathering students in one place is neither practical nor cost-effective. And even if this barrier can be overcome, reliable Internet access is even less common than computers, so coordinating and marking is a further challenge.

At the same time, South Africa takes part in the International Olympiad in Informatics (IOI), and we need a mechanism to select teams. We thus need to run a contest of comparable standard in order to select and train a team to represent South Africa at the IOI.

Today, the South African Computer Olympiad (SACO) features three rounds, described in detail in the following sections. This is followed by some statistics showing correlations of scores between rounds.

2. First Round

The first round is aimed at involving as many students as possible. It is a pen-and-paper round, similar to a mathematics olympiad, but with more focus on logic and programming. The question paper is mailed out to schools (via postal service, not e-mail) in advance, and teachers at the schools administer and mark the submissions. Answers are designed to be objective (often multiple choice or a number) rather than subjective (for example, an essay), so that teachers do not require any computer knowledge.

The round is offered in two divisions, junior and senior. The senior division is aimed to students in grades 10–12 (roughly 15–18 years old), while the junior division is restricted to students in grade 9 and lower. In the South African education system, subject choices are made when entering grade 10, so schools and students can use the results of the junior division to guide subject choices, while the senior division is helpful in making career choices.

The same paper is used for both divisions in this round and they are marked in the same way. The divisions are only distinguished when the students are ranked against one another. This is because having separate papers every year would add more difficulty in setting them, and in getting the teachers to photocopy and administer them. The paper is made to be like an aptitude test and the results are therefore valid for a range of ages – one just expects less from the average junior. Since a single paper is used for a wide age distribution and such a large variation in skill level within a division, the aim is to broaden the difficulty of the questions as much as possible.

Trying to gather and collate all the results from the hundreds of schools taking part would be an enormous task. Instead, certificates for the top three seniors and top three juniors are sent to each school, and results are not further compared. With enormous differences in education standards between advantaged and disadvantaged schools, a student who obtains 50% in a rural disadvantaged school probably has more potential than one who obtains 80% in an affluent urban school. The ranking within schools recognises this issue.

This format was first introduced in 2003, where it attracted 11 123 participants (South African Computer Olympiad, 2007). The junior division was added in 2006, and participation immediately increased to 31 926. In 2007, participation was 33 893.

3. Second Round

The second round of the SACO requires a computer. It is open to anyone, regardless of participation in the first round – this removes the need to ensure correct and consistent marking of the first round between schools.

The style of the problems is very loosely similar to the IOI, in that problems are algorithmic and intended to reach a specific answer, rather than testing the ability to build a user interface, database or other type of system. As with the first round, the objective nature of the answers makes it easier for teachers who have no experience in programming to mark solutions.

The input test data are included in the problem description, and students are required to submit both their source code and printouts of test runs on those test cases. The advantage of known data is that minimal work is required by teachers marking the paper, in that they do not need to compile or run submissions. An IOI-style automatic marking system is infeasible, as it would require Internet access, and would also cause difficulties for students not used to dealing with strict input and output formats or issues arising from differences between their local setup and the marking server. The main disadvantage of known data is that we usually have at least one test case that can be solved by hand, and some students tweak their programs until the desired answer is obtained without regard for the underlying bugs in their programs. Some students go further and simply hard-code answers into their code.

Because the results of the second round are used to select participants for the third round, the papers are re-marked centrally. Rather than re-mark all papers, schools are asked to send in their best result, and only asked for their second-best result when it is possible that the school may have two participants in the third round. Results are sent by postal service, and include the printouts of source code and test runs. The bulk of the points are awarded for producing correct output to the specified test cases. A small number of points are awarded for programming style, largely as a mechanism to break ties, but also to penalise solutions that have been written to solve only the test cases specified in the question.

An unfortunate consequence of using the postal service to gather results is that many students and teachers fail to follow the instructions, and by the time this is discovered it is too late to do anything about it. It is quite common to receive solutions that are missing either the source code or the sample run printouts, and these submissions are regrettably discarded.

As with the first round, a junior division, called *Start*, is offered to students in grade 10 and below. The change of age groups from the first round is due to programming being introduced in the second round. Schools do not teach any programming skills before grade 10 (and very few in grade 10). Trying to have a junior division limited to grades 9 and below would result in very few entries.

These students participate for enjoyment and experience, and are not eligible for the third round. Certificates are also sent to schools for the top three participants in each division. Unlike the first round, separate papers are set for the juniors and seniors, although some questions are shared between the divisions to reduce the amount of work required in problem-setting.

The second round dates back to 1987, when 1 750 students participated, with a similar format to today (although it was the first round until 2003, when the current first-round format was introduced). Perhaps surprisingly, participation has not grown steadily, but

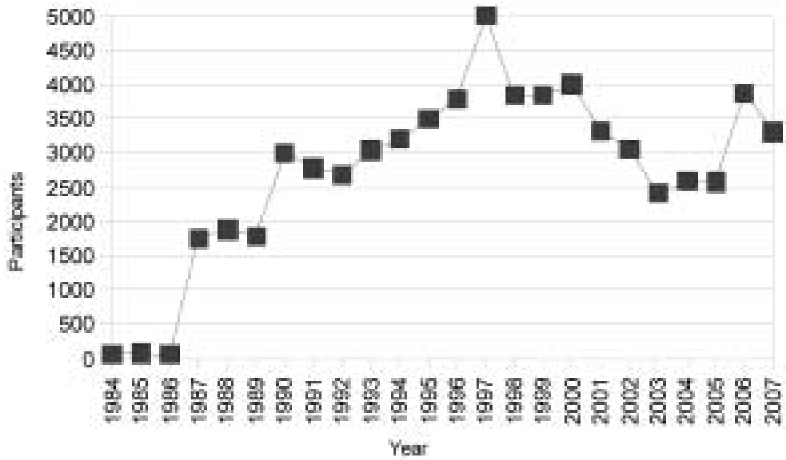


Fig. 1. SACO Second Round participation.

has varied considerably over time. The peak of 4 994 registrants occurred in 1997, when South Africa hosted the IOI, after which participation decreased. In 2003 the pen-and-paper first round was introduced, which reduced participation in the second round from 3 056 to 2 409. The junior division, introduced in 2006, has proved popular, with participation increasing to 3 873 that year. Fig. 1 shows participation in the second round since its inception.

4. Third Round

The best contestants from the second round are invited to participate in the third and final round of the SACO. The exact number of participants varies from year to year, with the aim of using a natural cut-off in the scores rather than forcing ties to be broken. Typically, 15 to 20 contestants are invited, but this number varies depending on funding.

The final round is an on-site event, hosted over the last ten years at the University of Cape Town (UCT). Students from outside Cape Town are provided with flights and accommodation, so cost does not prevent anyone from taking part in the final round. As most former IOI participants in South Africa carry on to study at UCT, there is never a shortage of on-site judging staff.

As the final round contains problems that are at a level far greater than the students encounter at school, they are sent training material prior to the final round to help them prepare. This includes printed copies of previous final round papers and pointers to various online resources. Due to participants being spread across the country, personal training cannot be provided and they therefore often resort to self-training.

The competition format of the final round follows that of the IOI quite closely, and new trends in task descriptions, types, compilers and so on are quickly adopted. As with the IOI, the contest consists of two days, each with a five hour contest featuring three problems, and with a similar automated online grading system. While the problems are similar in style to the IOI, they are of slightly lower difficulty so that all the contestants are able to at least attempt them.

The only major technical difference from the IOI is in the languages offered: the IOI-standard languages of C, C++ and Pascal are available, but Java and more recently Python are also provided. Java is the main language taught in South African schools; Python was added due to the backing of a sponsor that wished to grow the language in South Africa, and it has proven extremely popular as it is easy to learn and offers powerful features not easily accessible in Java. In fact, although the top six Python users in the second round are guaranteed a place in the final (so that six Python prizes can be awarded – see below), these top six have always done well enough to earn an invitation without this provision.

We have found that although Java programs are usually somewhat slower than equivalent C programs (a frequent objection whenever a proposal is made to introduce Java to the IOI), the speed is sufficiently comparable that we can use the same time limits for Java as the other languages. Python, on the other hand, is a scripting language and is 1–2 orders of magnitude slower than the other languages. We have thus implemented different time limits for Python. The ratio of time limits between Python and other languages is reviewed each year based on the performance of reference solutions. In the 2007 SACO, the ratio used was 10.

At the IOI, the afternoons after the contests are free time for the contestants to review their scores and make appeals. At the SACO, this time is somewhat more structured. The judges lead a discussion of proposed solution methods (often leading out of more informal discussion during the lunch break), and after the first day of competition there is commonly some training on general topics.

The SACO offers prizes, and winners are often offered scholarships and bursaries, so the judging is as strict and impartial as the IOI, with no opportunity to “just fix one bug” or “just correct the file name”. Unfortunately, this also limits the degree to which it can be used as a training opportunity, because we are unable to help with programming questions during the contest, and also cannot provide one-on-one help beforehand. Since the introduction of junior divisions into the earlier rounds, we have also had a semi-official junior division of the final round, to which a few (around six) top-performing juniors from the second round are invited. As this is a for-fun event with no cash prizes, the judges are free to provide hints and advice during the contest, and this forms a valuable learning experience for the contestants. We believe this approach has been successful, with several junior contestants returning as regular final-round contestants in later years.

Table 1 provides statistics on the language usage at the final round over the past. Before Python was introduced, the majority of students used Pascal and Java, the languages taught at schools. The gradual increase in Java usage corresponds to the increased number of schools moving from Pascal to Java as a teaching language. The small number of C++ users are typically former IOI participants who were required to learn either C++ or Pascal for the IOI.

The sudden shift to Python upon its introduction in 2005 is immediately evident from the data. This is mostly due to the sponsorship of cash prizes for the top Python users, which are significantly larger than the standard prizes.

Table 1
Language distribution in the third round, 2002–2007

	C/C++	Pascal	Java	Python
2002	0 (0%)	8 (80%)	2 (20%)	
2003	3 (25%)	6 (50%)	3 (25%)	
2004	3 (25%)	3 (25%)	6 (50%)	
2005	0 (0%)	5 (21%)	4 (17%)	15 (63%)
2006	0 (0%)	0 (0%)	3 (20%)	12 (80%)
2007	0 (0%)	1 (5%)	6 (29%)	14 (67%)

5. Statistics

We have collected a limited set of data from the 2007 contest, consisting of the scores for student whose second-round papers were centrally graded. Since this is only done where there is a chance that a paper is within the top hundred schools (for example, submissions that only attempted one question are not graded), this is not a statistically random sample. Nevertheless, some interesting results can be obtained from this data.

Fig. 2 shows a scatter-plot of scores in rounds 1 and 2, for seniors in round 2. It should be noted that of the 126 students for whom data was captured, only 76 entered the first round, and only those results are shown in the figure (the reason for this is not known, but some students with the ability to do well in the second round may not consider the first round sufficiently challenging to interest them). The triangular shape of the figure is interesting: it suggests a high score in the first round indicates a capability with problem-solving, but that this not does always translate into the ability to implement solutions. Many students in South Africa do not have access to a programming course at high school, and potentially talent is being wasted due to lack of education.

We also computed the correlation of the scores between these rounds. The Pearson product-moment correlation coefficient was 0.369. Under the assumption of a normal distribution, this would be highly statistically significant, but this is not necessarily a valid assumption as capturing results only for top students will skew the distribution. The Kendall tau (Kendall, 1938) value (a non-parametric measure of correlation) is 0.245, and this is also highly statistically significant ($p = 0.0024$).

Fig. 3 shows the same comparison between the second and third rounds. Here the data is complete, as we have second and third round results for all 22 participants in the third round. The graph suggests that a very strong performance in the second round is a good indicator of a strong performance in the third round, but that a weaker performance in the second round is not necessarily indicative of performance in the third round.

The Pearson's product-moment correlation coefficient in this case is 0.54, and is highly statistically significant ($p < 0.01$), although again this may not be statistically valid as the second round performances will not be normally distributed. The Kendall tau value is 0.31 and the p-value is roughly 0.05; R (R Development Core Team, 2007) warns that it cannot compute an exact value when there are ties.

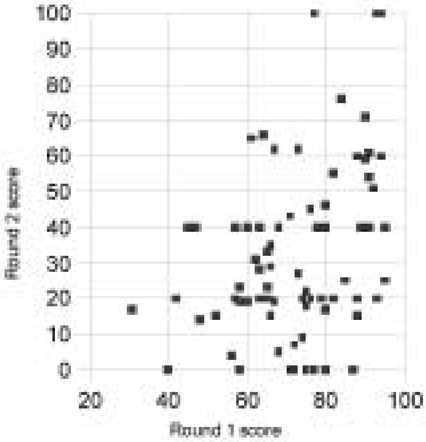


Fig. 2. Scores in round 2 against round 1.

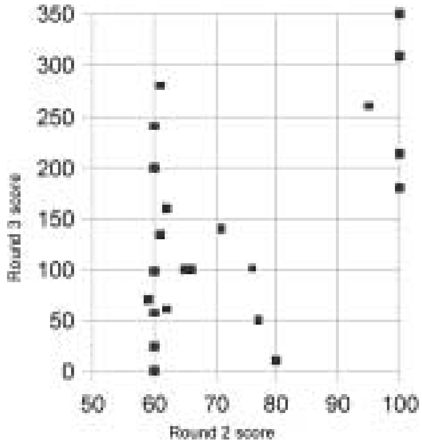


Fig. 3. Scores in round 3 against round 2.

6. Conclusions

In spite of the limited ICT infrastructure in South Africa and South African schools, we are able to run a large and successful computer olympiad that hopefully encourages students to pursue careers in computing. This is achieved by limiting the use of technology in each round to what is generally available. The first round requires no computer, and so it is accessible to all students even though few schools have computers. Our statistical analysis also shows that the first round has provided a more accessible medium for students with strong problem-solving abilities, who have not yet developed the skills to master programmatic problem-solving.

Of course, a programming contest should not be run completely without computers, and they are required for the second round. However, we use printouts rather than internet access for submission, and attempt to keep marking as simple as possible so that participation is possible even when there is no trained computer studies teacher at the school. We can also observe from Fig. 2 that students who perform well in the second round, have similarly strong performances in the first round – indicating overall strong problem-solving abilities.

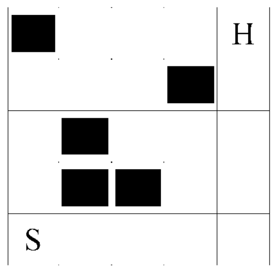
By the final round, we can provide an experience similar to the IOI, as the small number of contestants affords us the ability to bring them all to a single site and use a web interface on the local network. Fig. 3 shows the value of this round in selecting an IOI team: while there is some correlation between second- and third-round performances, many of the finalists had similar scores in the second round (around 60), but could be separated by the more challenging conditions of the final.

7. Appendix: Sample Problems

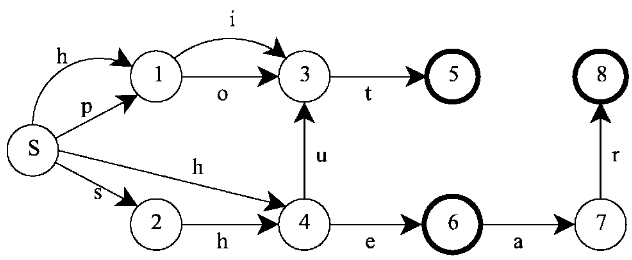
7.1. *First Round*

The following are samples of easy, medium and difficult problems from the 2006 Olympiad:

1. Imagine a country called SACO that uses 5c and 7c coins. Which of the following amounts cannot be paid using only 5c and 7c coins?
   1. 27
   2. 26
   3. 24
   4. 23
2. Sally (S) wants to go home (H). She can only move up or right one square each time. She is not allowed to go through black squares. How many paths can she pick from to go home?



3. A finite-state machine (FSM) is a . . . (explanation of an FSM follows). What words does the following FSM recognise?



7.2. *Second Round*

The following is a sample question taken from the 2007 Olympiad.

Description

Strings are just a series of characters “strung” or joined together. Substrings are strings that are, in fact, just a part of a larger string. One might, for various reasons, wish to find if a string is merely a substring of another string, sometimes disregarding such things as case (UPPER and lower) or punctuation.

Task

Your task is to write a program that finds and prints all occurrences of a word (substring) within a piece of text. This word may be hidden, it may contain spaces or punctuation, and it might appear with different capitalization. The program must accept 2 strings, the first being the main string, and the second the substring that is to be searched for in the main string. If no substrings are found, “No strings found” must be printed.

Constraints

The length of each string will be $<$ 255 characters.

Sample Run

Input

```
It's behind the intercom. Put erasers to one side computer
```

Output

```
com. Put er
```

Test your program with

```
This suit is black!!
not

"You thought your secrets were safe. You were wrong." - Hackers
gh

Donald likes Mall shops where he and his friends discuss
idealism all day long.
Small
```

7.3. *Third Round*

Third round questions are typical of the IOI and similar contests, so we do not include any samples here.

References

South African Computer Olympiad (2007). *History and Statistics – South African Computer Olympiad.* Retrieved 9 March 2008 from `http://www.olympiad.org.za/history.htm`

R Development Core Team (2007). *R*: *A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. `http://www.R-project.org`

Kendall, M. (1938). A new measure of rank correlation. *Biometrika*, **30**, 81–89.

B. Merry took part in the SACO from 1995 to 2000. Since then he has been involved in running the contest, as well as South Africa's IOI training program. He obtained his PhD in computer science from the University of Cape Town (UCT) and is now working as a software engineer at ARM.

M. Gallotta is the deputy coach of the SA IOI team. After participating in the IOI in 2004, Marco is currently in charge of task selection and the technical side of running the SACO, as well as being the coach of the ACM ICPC team at UCT. Having recently obtained a BSc (Hons) in computer science at UCT, he is now there temporarily as a research assistant working on RoboCup.

C. Hultquist has been involved with the SACO since 1995, when he first started taking part as a contestant. Upon commencing his studies at UCT, he became involved in the running of the contest and the training of South Africa's IOI team. He recently started working for D.E. Shaw & Co. (U.K.), Ltd, and is also putting the finishing touches to his PhD in computer science.