Encouraging algorithmic thinking without a computer

Benjamin Burton

The University of Queensland, Australia

IOI Conference 2010

Our problem

IOI and other informatics competitions are increasingly popular.

But . . .

In Australia, we struggle to find secondary school students.

- Entry-level mathematics contest: historically $\sim 500\,000$
- Entry-level programming contest: historically < 150

Reports from other countries show we are not alone.

The reasons

Why so few students?

- Curriculum:
  Very little computer programming or algorithm design at secondary schools.
  - Few students with the right skills
  - Teachers do not know who they are
- Technology:
  Programming contests require a dedicated computer.
  - Difficult to schedule and supervise
  - Severely limits the number of entrants per school
- Grading:
  You cannot score if you cannot produce a working program!
  - Many low or zero scores
  - Discouraging and disheartening to inexperienced students

A natural solution

What about a pen-and-paper competition?

- Multiple choice or short answer tasks
- Quick to solve (correctly or incorrectly)
- Highly approachable even with no prior knowledge
- Range in difficulty from challenging to very easy

A natural solution . . . that is spreading

Several countries are doing this already.

Bulgaria (1992–)

- Mixed into the mathematics contest Chernorizets Hrabar

South Africa (2003–)

- See Bruce Merry's paper (IOI conference 2008)

Lithuania et al. (2004–)

- Bebras: Uses computers, but in a much more accessible way

Australia (2005–)

- Australian Informatics Competition: Described in this talk

. . . and more!

The main difficulty

How to retain a focus on algorithmic thinking?

This is a core aim of the Australian Informatics Competition (AIC).

Especially difficult for multiple choice:

- No prior knowledge (no pseudocode, no programming)
- Not just traditional mathematical problems and puzzles

The remainder of this talk:

- Introduce the AIC
- Show how it achieves this algorithmic focus

The Australian Informatics Competition

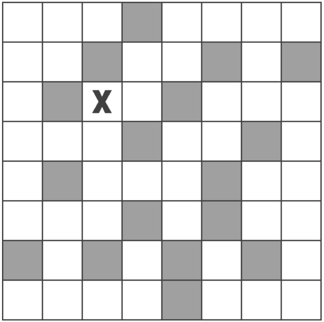
15 questions in one hour:

- 6 multiple choice (A–E)
- 9 integer answers (0–999), used for three-stage tasks

Benefits:

- Easy to administer for teachers
- Accessible and engaging for students
- Easy to grade for administrators

Example multiple choice task: Dungeon (AIC 2005)



You are moving around the grid, starting at **X**.

A *move* takes you through any number of white squares in a single direction.

What is the smallest number of moves in which you can reach *any* white square?

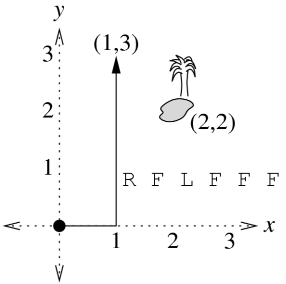
(A) 8 (B) 9 (C) 10 (D) 11 (E) 12

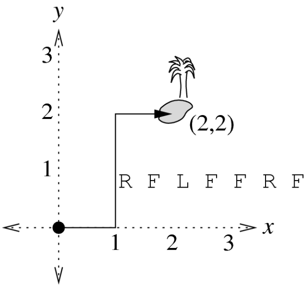
Example three-stage task: Lost (AIC 2007)

From $(0,0)$ facing north, you have directions:

- **F** means step forward one unit;
- **L** / **R** means turn 90° to the left / right.

You are aiming for an oasis, but one **R** has been deleted from your list. After how many letters should the **R** be inserted?





1. R F F L F L F F F R F F
   Oasis at $(3,3)$
2. R F F L F R F F L F F L F L F R F F R
   F F L F L F R F R F F R F L F F L F
   Oasis at $(5,5)$
3. R F F F L F F R F F F R F R F F R F F F F
   F F L F F L F F F L F L F F F F F L F F
   Oasis at $(8,8)$

Types of tasks: Multiple choice

Puzzle-based tasks that target various aspects of algorithms.

- Tracing tasks:
  Easy tasks that ask students to follow instructions
- Algorithmic tasks:
  Encourage students to develop an “informal algorithm”
- Logic tasks:
  Non-algorithm puzzles that focus on rigour and case analysis
- Analysis tasks:
  Probe the behaviour of a given algorithm or problem

The trouble with multiple choice

Multiple choice tasks are small, with just one “test case”.

Difficult to encourage students to develop algorithms:

- Tasks are easy to attack with educated guesswork

Difficult to evaluate whether student have found an algorithm:

- Educated guesswork often gets the right answer!

This is why the AIC includes three-stage tasks.

Three-stage tasks

Recall the example *Lost* from before.

Three related questions with integer answers (0–999).

All pose the same problem, but with “data sets” of increasing size:

- Set #1 is small, and can be solve ad-hoc
- Set #2 is larger, and helps students develop systematic techniques
- Set #3 is even larger, and students can apply these systematic techniques quickly and efficiently

$\Rightarrow$ Student effectively “develop an algorithm”!

This both encourages and evaluates algorithm development.

How well is it received?

Largest-ever programming contest: **142 students**

First-ever AIC: **2511 students**

Now well over 3000 with entrants from New Zealand and Singapore.

Our new challenge: To nurture and encourage students who enjoy the AIC, and to show them what programming and algorithms are all about.

See the full paper for many more examples of tasks.