Problem setting in mathematics and informatics

Weaving proof into programming

Benjamin Burton

The University of Queensland, Australia

WFNMC Congress 2010

What is IOI?

The International Olympiad in Informatics (IOI):

- Second-largest of the Science Olympiads
- Focus on correct and efficient algorithms
- Students submit computer programs, not written solutions

Algorithm design has a rich interplay with mathematics.

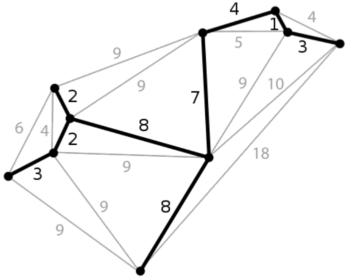
Bedtime reading:

> *Donald E. Knuth:*
> *The Art of Computer Programming (vols. 1–3)*

How does proof feature in the IOI?

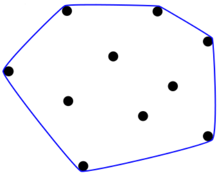
Proof of correctness:

- Example: **Minimal spanning tree**
- A correct algorithm just repeatedly chooses the shortest “legal” edge remaining (Kruskal’s algorithm).
- It is interesting (and non-trivial) to prove that this is minimal.



Proof of complexity:

- Example: **Graham scan for 2-D convex hulls**
- This algorithm “looks” like $O(n^2)$ running time, but you can prove that bad cases are “rare”, giving $O(n \log n)$ instead.



Problem setting in informatics olympiads

Ordinary problems:

- Think of a problem to fit a topic or style of solution (graph theory, dynamic programming, combinatorics, . . .)
- Often produces “standard” (non-exciting) problems

Fresh, interesting problems:

- Think of a real-world problem: ”How would I . . . ?”
- *Disadvantage:* You must solve it yourself without any hints
- *Advantage:* Solutions can be highly original and creative
- *Caveat:* Real-world problems are often infeasible (NP-hard)
  $\longrightarrow$ Find ways to simplify the problem

The real-world method is time-consuming but often highly rewarding.

Example: Citizenship

From 2006 Australian team selection exam
(problem by Bernard Blackham)

Origin

(in the pub): “Wouldn’t it be cool to gain citizenship of as many countries as possible?”

Example: Citizenship

Final task

Countries have rules for gaining and losing citizenship:

Country	*Years present to gain*	*Years absent to lose*
Australia	2	15
Burgmanistan	6	3
France	6	11
Italy	4	7
Ursulia	8	5

You can fly infinitely fast. What is the largest number of simultaneous citizenships you can gain?

Input: Rules for $\leq 100\,000$ countries

Output: Number of citizenships (example solution: 4)

Running time: 1 second limit $\longrightarrow$ aiming for $O(n \log n)$

Ideas for solutions

Sort descending by years to lose?

Sort descending by years to gain?

Sort some other way...?

Can we visualise the problem?

How to turn these ideas into a provably correct algorithm?

Evaluating IOI solutions

IOI solutions are evaluated by behaviour:

- Students submit code (C++, Pascal, etc.), not written proofs
- Judges prepare a thorough set of official test data
- Students gain points for every data set they solve correctly

Advantages:

- Students cannot “fudge” the details
- Allows feedback during the exam, live scoring, ...

Disadvantages:

- Automated evaluation can be unforgiving
- How to distinguish between a proof and a good guess?

Guessing in the IOI

Most "obvious guesses" for algorithmic problems are wrong . . .

. . . but sometimes they work (e.g., minimum spanning tree)!

When "easily guessable" algorithms do work:

- Proofs often become interesting
- Evaluation often becomes difficult

The risk: penalising good students who "waste" time proving their algorithms correct

How to encourage proof?

Establish a history of making the "easily guessable" solutions wrong:

- Maybe not all the time, but certainly most of the time

Maintain a strong culture of proof and critical thinking during training camps:

- Invite students to "break" each others' algorithms, or prove why they can't
- Hold events that focus on proofs and counterexamples (e.g., Australian codebreaker contests)

Want to know more?

My “introduction to IOI” from the WFNMC Congress 2006:

> *Informatics olympiads: Approaching mathematics through code, Mathematics Competitions* ***20*** *(2007), no. 2, 29–51*

More on IOI problem setting:

> *Creating informatics olympiad tasks: Exploring the black art (B.B. & M. Hiron), Olympiads in Informatics* ***2*** *(2008), 16–36*

IOI website (includes past papers):

> `http://www.ioinformatics.org/`