Breaking the Routine: Events to Complement Informatics Olympiad Training

Benjamin Burton (Australia)

IOI 2008

Overview

Training schools in Australia are built around *lectures*, *laboratories*, *contests* and *task discussions*.

Here we focus on additional “extra” events:

- *Codebreakers* (finding counterexamples)
- *B-sessions* (focus on implementation)
- *Team events* (teamwork and puzzles)

Codebreakers

Motivation:

- Most discussions focus on *correct* algorithms
- Students should be able to identify *incorrect* algorithms
- Students should be able to generate pathological test cases

Structure:

- Online contest with live scoreboard
- Students receive ~ 3 tasks, ~ 15 bad solutions
- Students submit input files to break these solutions

Scoring:

- $+10$ if the input file breaks the solution
- -1 if the input file does not break the solution
- -2 if the input file is invalid
- Unlimited number of attempts allowed

Codebreakers: Example

Codebreaker Submissions for Bernard Blackham

Log out

Below is a list of all problems and source files for this codebreaker.

Problem	Source	Score out of 10	Attempts	Actions
1. Wetlands	wetlands1.java	10 (success!)	1	View submissions
	wetlands2.c	10 (success!)	1	View submissions
	wetlands3.cpp	8 (success!)	2	View submissions
2. Mansion	mansion1.cpp			Break!
	mansion2.cpp	8 (success!)	2	View submissions
	mansion3.cpp			Break!
	mansion4.c	10 (success!)	1	View submissions
3. Invasion	invasion1.cpp	8 (success!)	2	View submissions
	invasion2.cpp	-1	1	Break!
	invasion3.cpp			Break!
	invasion4.pas			Break!
	invasion5.cpp			Break!
4. Restaurants	restaurant1.cc	9 (success!)	2	View submissions
	restaurant2.py	-3	3	Break!
	restaurant3.cpp	10 (success!)	1	View submissions
	restaurant4.cc	10 (success!)	1	View submissions

Your total score for the codebreaker so far is 79.

Codebreakers: Selecting Tasks and Solutions

Tasks:

- Students already know the tasks
- Often taken from a recent national contest

Bad solutions:

- Code is clear and readable (not obfuscated)
- Different solutions fail in different scenarios

Types of errors:

- Implementation errors (e.g., $<$ becomes $\leq$)
- Algorithmic errors (e.g., incorrect greedy solution)
- Comprehension errors (solves a slightly different task)

B-Sessions

Motivation:

- Many students sloppy with implementation
- Some students afraid to code up complex algorithms
- Some students reluctant to attempt difficult tasks

Structure:

- One difficult problem is handed out the day before
- Students meet for one hour to discuss the solution
- Students implement this solution in $2\frac{1}{2}$ hours, exam conditions

Clear separation of *algorithms* and *implementation*.

Team Events

Motivation:

- To encourage teamwork
- To expose students to other areas in computer science
- To spend an afternoon doing something noisy and fun!

Structure:

- Choose a topic (e.g., classical cryptography and cryptanalysis)
- Begin with a one-hour lecture on this topic
- Teams are given a series of puzzles to solve in the lab (e.g., coded messages to decrypt)
- The final puzzle shows the location of a hidden prize

Conclusion

Different types of events can work well in training schools:

- Break the usual routine
- Develop different skills
- Introduce students to new topics

Conference paper contains details and technical requirements.

More suggestions are welcome!