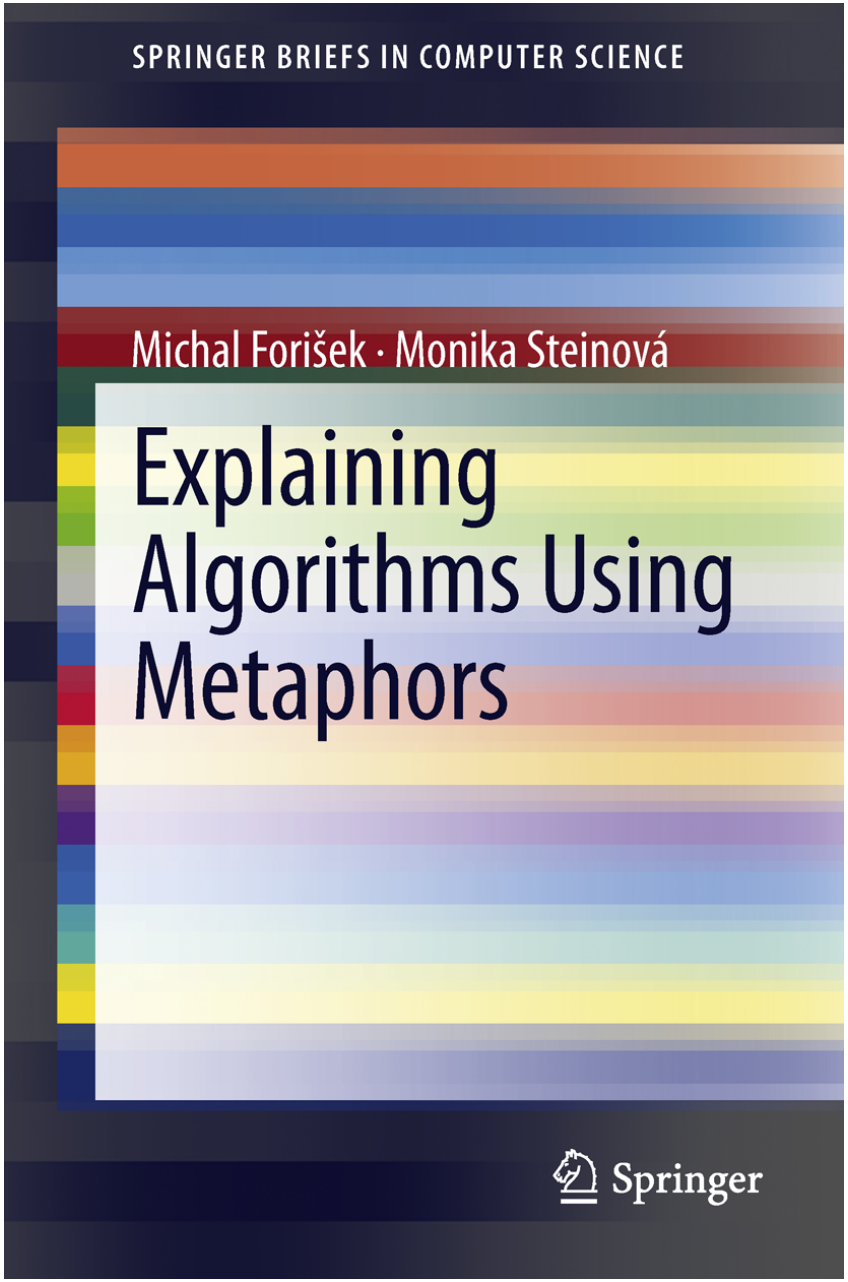
Pushing the Boundaries of Programming Contests

Michal ‘mišof’ Forišek

Comenius University, Bratislava, Slovakia

IOI 2013, Brisbane, Australia


SPRINGER BRIEFS IN COMPUTER SCIENCE
Michal Forišek · Monika Steinová
Explaining Algorithms Using Metaphors
Springer

Internet Problem Solving Contest

- `http://ipsc.ksp.sk/`
- annual contest since 1999 (held in May/June)
- last year: 1337 teams / 87 countries
- challenging problems, skilled contestants
- separate ranklists for secondary schools
- problem solving as the goal (programming=tool)
- always going beyond algorithmic batch tasks

Crypto-related tasks

- IPSC 2013 I: restore encrypted images
- IPSC 2012 I: crypto: recover passwords from their hashes
- IPSC 2012 K: keys and locks
- IPSC 2011 L: a real-time man-in-the-middle attack

Non-traditional computation models

- separate algorithmic thinking from language proficiency
- bring the contestants outside of their comfort zone

Models used in the past

- many-one and Turing reductions (2013),
- regexes (2013),
- log-space programs (2012),
- Fractran (2011),
- alternating machines (2005),
- reversible programs (2002),
- Wang tiles (2001),
- ... and many more.

Non-traditional computation model: Fractran

Fractran

- the state is a positive integer s
- the program is a sequence of positive fractions
- computation step:
  find the first fraction f such that sf is an integer
  change s to sf

Sample task

The initial state is a number $s = 2^x 3^y 5$.
Write a Fractran program that terminates with the value $s = 5$ whenever $x = y$, or the value $s = 7$ whenever $x \neq y$.

$$\left(\frac{1}{6}, \frac{7}{10}, \frac{7}{15}, \frac{1}{2}, \frac{1}{3} \right)$$

Non-traditional computation model: Fractran

Fractran

- the state is a positive integer s
- the program is a sequence of positive fractions
- computation step:
  find the first fraction f such that sf is an integer
  change s to sf

Sample task

The initial state is a number $s = 2^x 3^y 5$.
Write a Fractran program that terminates with the value $s = 5$ whenever $x = y$, or the value $s = 7$ whenever $x \neq y$.

$$\left(\frac{1}{6}, \frac{7}{10}, \frac{7}{15}, \frac{1}{2}, \frac{1}{3} \right)$$

Non-traditional computation model: Fractran

Fractran

- the state is a positive integer s
- the program is a sequence of positive fractions
- computation step:
  find the first fraction f such that sf is an integer
  change s to sf

Sample task

The initial state is a number $s = 2^x 3^y 5$.
Write a Fractran program that terminates with the value $s = 5$ whenever $x = y$, or the value $s = 7$ whenever $x \neq y$.

$$\left(\frac{1}{6}, \frac{7}{10}, \frac{7}{15}, \frac{1}{2}, \frac{1}{3} \right)$$

Code analysis tasks

A less-traditional one:

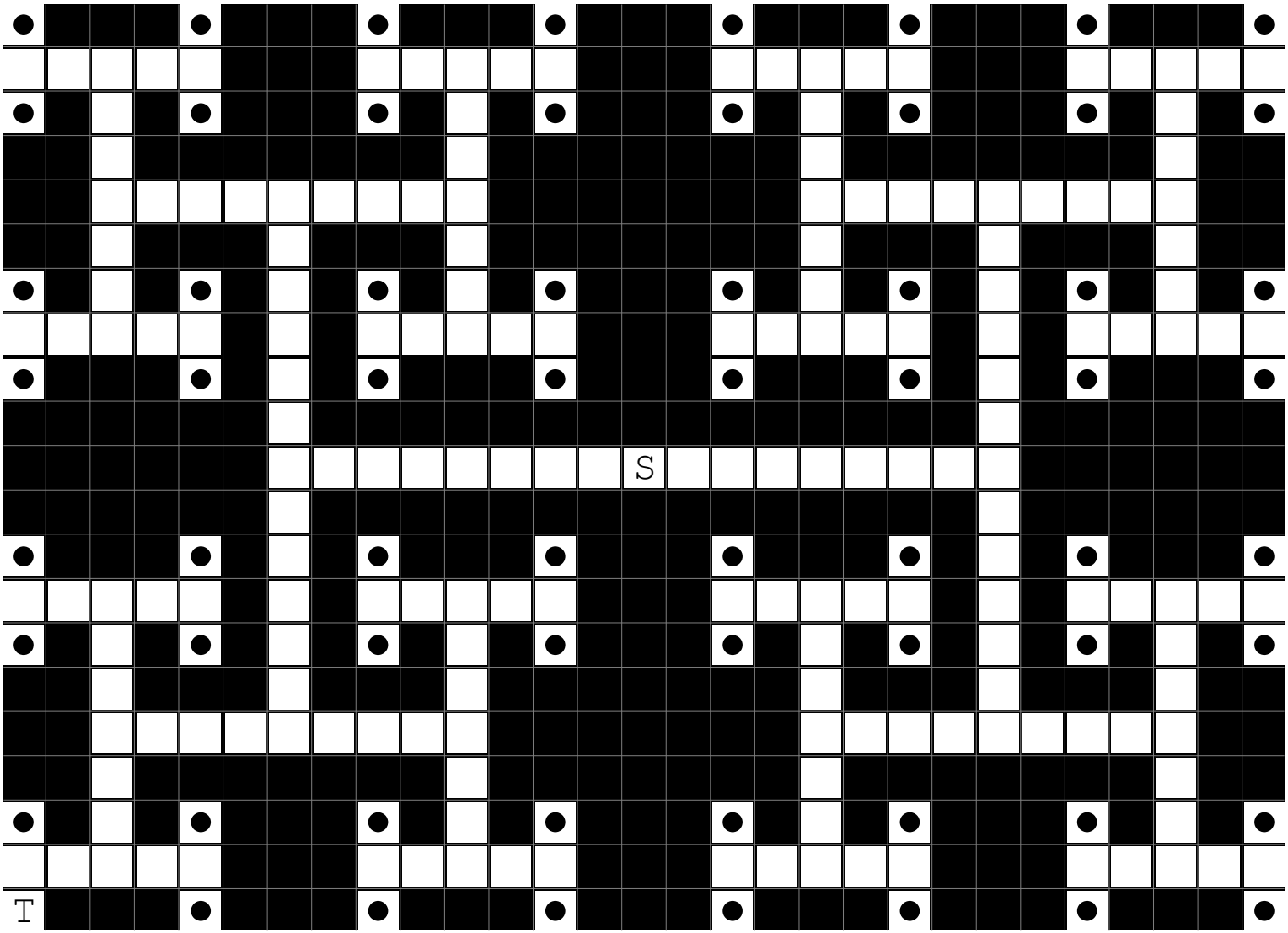
IPSC 2011 B: BFS killer

Given is a simple code that runs BFS on a grid with obstacles.
Find an input such that the queue will overflow.

A common misconception:
on an $n \times n$ grid the BFS queue size is always $O(n)$.

(Even occured in an IOI task submission.)

Code analysis tasks



Low-level language task

IPSC 2008 C: Comparison mysteries

Declare a numeric variable `x` and initialize it to a non-zero value.
Your variable must then satisfy `x == -x`.

Declare three numeric variables `x`, `y`, and `z`.
Initialize them to any values.
Your variables must then satisfy `x == y && y == z`.
However, they must NOT satisfy `x == z`.

```
int x = -2147483648;
```

```
int x   = 1234567890;
float y = 1234567890;
int z   = 1234567891;
```

Low-level language task

IPSC 2008 C: Comparison mysteries

Declare a numeric variable `x` and initialize it to a non-zero value.
Your variable must then satisfy `x == -x`.

Declare three numeric variables `x`, `y`, and `z`.
Initialize them to any values.
Your variables must then satisfy `x == y && y == z`.
However, they must NOT satisfy `x == z`.

```
int x = -2147483648;
```

```
int x   = 1234567890;
float y = 1234567890;
int z   = 1234567891;
```

Low-level language task

IPSC 2008 C: Comparison mysteries

Declare a numeric variable `x` and initialize it to a non-zero value.
Your variable must then satisfy `x == -x`.

Declare three numeric variables `x`, `y`, and `z`.
Initialize them to any values.
Your variables must then satisfy `x == y && y == z`.
However, they must NOT satisfy `x == z`.

```
int x = -2147483648;
```

```
int x   = 1234567890;
float y = 1234567890;
int z   = 1234567891;
```

“Lateral thinking” task #1

Alpha Centauri Tennis

tennis generalized to n players
statement: detailed rules on how games and sets are scored
input: a match record: for each point the person who won
example: AABABCAAAABABABB...

input is guaranteed to be a valid match record
task: find out who won

The winner is the one who won the last ball.

Output the last character of the string.

“Lateral thinking” task #1

Alpha Centauri Tennis

tennis generalized to n players
statement: detailed rules on how games and sets are scored
input: a match record: for each point the person who won
example: AABABCAAAABABABB...

input is guaranteed to be a valid match record
task: find out who won

The winner is the one who won the last ball.

Output the last character of the string.

“Lateral thinking” task #2

Know Your Crypto

statement summary:

– sorry, this task is impossible to solve

– it was prepared at the last possible moment

– we just took the message and randomly scrambled each letter

– here is the code we used: `srand(time(0)); ...`

Solution: try all possible return values of `time(0)`.

real issue: e.g., Casino de Montréal in 1994.

“Lateral thinking” task #2

Know Your Crypto

statement summary:
– sorry, this task is impossible to solve
– it was prepared at the last possible moment
– we just took the message and randomly scrambled each letter
– here is the code we used: `srand(time(0)); ...`

Solution: try all possible return values of `time(0)`.

real issue: e.g., Casino de Montréal in 1994.

“Lateral thinking” task #3: IPSC 2012 B

“Lateral thinking” task #3: IPSC 2012 B

A clear world record: the shortest statement!

Task still solvable: use the error messages from the grader.

Usual response to the first submission:
“`Wrong answer:  Not a sequence of positive integers!`”.

That's all, folks!

- read the paper for more tasks
- visit the IPSC website for even more tasks
- feel free to reuse them!
- ask me questions (now or anytime later)

- thanks for your attention!