The difficulty of programming contests increases

Michal Forišek

Department of Theoretical Computer Science
Faculty of Mathematics, Physics and Informatics
Comenius University
Bratislava, Slovakia

January 13, 2010

Goals and motivation

What do we want?

Research the difficulty of programming contests.

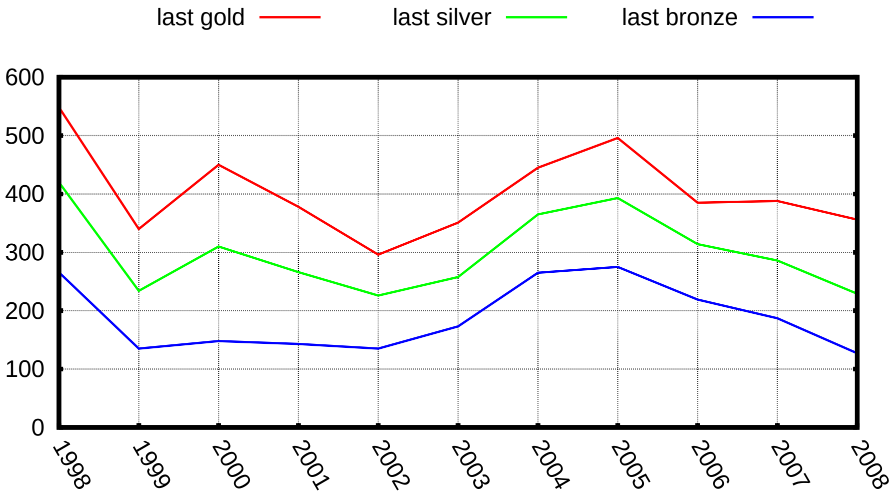
Why bother?

Better understanding helps us do a better job.

Is this even worth researching?

Intuition is easy, proving it may be surprisingly hard.
See motivational example that follows.

IOI Medal Boundaries



Prior research

Manual task classification

Skiena and Revilla (2003)

Programming challenges

Kiryukhin and Okulov (2007)

Methods of Problem Solving in Informatics

Verhoeff (2009)

20 Years of IOI Competition Tasks

Item Response Theory

Kemkes, Vasiga, Cormack (2006)

Objective Scoring for Computing Competition Tasks

Forišek (2009)

Using Item Response Theory to Rate (Not Only) Programmers

Four main results

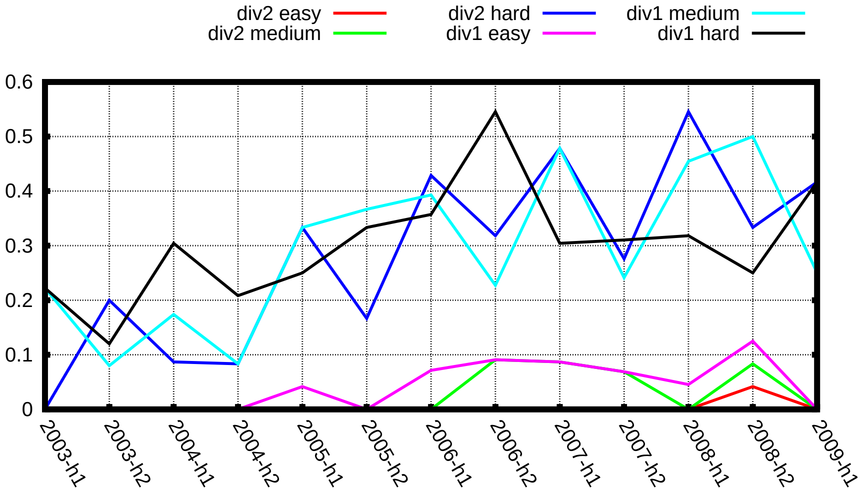
1. The set of topics is growing.
2. The topics previously considered difficult now appear early.
3. **The difficulty of programming contest tasks increases.**
4. The skills of (both top and average) contestants increase.

Survey of NWERC median tasks

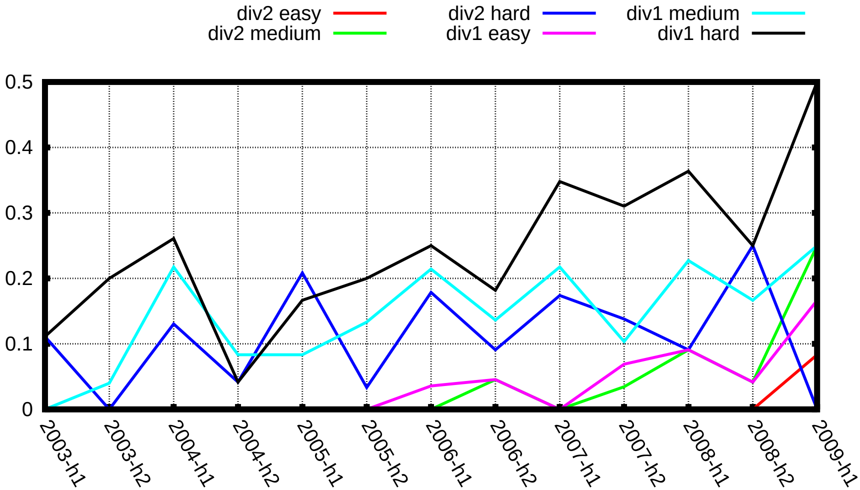
respondents: 33 contestants from more than 20 countries
question: order a subset of these tasks according to difficulty

	0	1	2	3	4	5	6	7	8	9	10	11
internet/0		0	0	1	2	0	0	1	0	2	0	0
space/1	27		20	11	18	8	8	5	3	12	5	5
papergirl/2	23	5		8	11	6	6	5	3	8	4	6
railroads/3	21	14	16		19	12	12	9	6	8	6	8
dates/4	24	6	15	6		8	7	6	4	9	5	5
floors/5	24	15	15	12	17		10	7	7	13	6	8
boss/6	23	14	15	10	13	12		4	5	9	6	8
taxicab/7	22	17	15	15	18	14	15		15	13	14	13
tantrix/8	24	20	21	18	22	15	16	8		19	13	11
setstack/9	21	10	13	13	18	10	11	7	5		10	9
escape/10	26	20	17	16	20	15	15	7	11	13		14
mobile/11	22	15	16	13	18	12	12	7	10	12	7	

Dynamic Programming at TopCoder



Graph Theory at TopCoder



ACM ICPC: towards faster algorithms

Single-source shortest paths (Dijkstra's algorithm):

date	N	M	comment
2002-04-20	1 000		requires preprocessing
2002-07-27	200		2nd shortest walk
2004-10-16	1 000	10 000	k shortest walks
2005-09-24	1 000		number of shortest paths
2006-01-21	20 000	50 000	
2007-12-01	100 000	1 000 000	
2009-07-18	10 000	100 000	additional complications

TopCoder: towards more and faster solvers

Division 1 easy task (evaluating voting ballots)

year	solved	best time	avg time
2003	85/160 (53.13%)	0:07:07	0:23:01
2008	469/583 (80.45%)	0:02:48	0:16:59

Division 1 hard task (max flow/min cut)

year	solved	best time	avg time
2003	11/138 (7.97%)	0:21:45	0:35:51
2007	102/385 (26.49%)	0:02:44	0:18:13

Conclusions and questions

- Tasks are getting harder, contestants are getting better.
- Clearly, one influences the other.
  A virtuous circle or a vicious one?
- When and how will the process stop?
- Should we attempt to influence it?

Answering these needs much more research,
we only made the first few steps here.