# c0d1l1ty

## Application of Olympiad-Style Code Assessment to Pre-Hire Screening of Programmers

Grzegorz Jakacki  Marcin Kubica  Tomasz Waleń

**IOI 2011**

# Outline

Outline:

- History
- Tasks and Task Preparation
- Codility Web Service
- Evaluation Technology
- Correctness and Soundness
- Scoring
- Conclusions

# History

Exoweb — the beginning:

- Exoweb's demand for new employees (2005)
- Crucial ability to write correct code
- Hires-to-candidates ratio $\approx$ 1 : 50
- Average interview time of 100 min.
- 80 h of senior engineer's time per one hire
- Main cause of rejection — inability to write bug-free code

# History

Solution:

- Screening — programming test
- Automated evaluation
  Exobench — command-line evaluation system
- 90 % screening accuracy
- 12 h of senior engineer's time per one hire
- Post-screening hires-to-candidates ratio $\approx$ 1 : 5
- Codility is born (2008)
- Reimplementation of pre-hire screening as a web service.

# Tasks

Task characteristics:

- Goal of IOI and other programming contests:
  select top contestants and fairly distribute points among other contestants
- Goal of pre-hire screening:
  eliminate candidates that should not be employed
- Binary outcome of pre-hire screening —
  reject or send for interview
- Tasks are much more elementary (compared to IOI)

# Task Preparation

Task preparation:

- Over a hundred tasks (and growing)
- General task preparation schema is similar to IOI
- Simpler task formulation
- No narrative story
- Only batch tasks are supported

# Task Preparation

Task preparation:

- Procedural interface (instead of standard I/O)
- Testing procedures instead of predetermined test files
- Tests profile similar to IOI:
  - correctness tests of various sizes,
  - border-cases tests,
  - efficiency tests

# Task Preparation for Multiple Programming Languages

Variety of programming languages:

- C, C++, C#, Java, JavaScript, Pascal, Python, Perl, PHP, Ruby
  (and still growing)
- Fully compilable, byte-code level compilable and scripting languages
- How to deal with such a variety of programming languages?

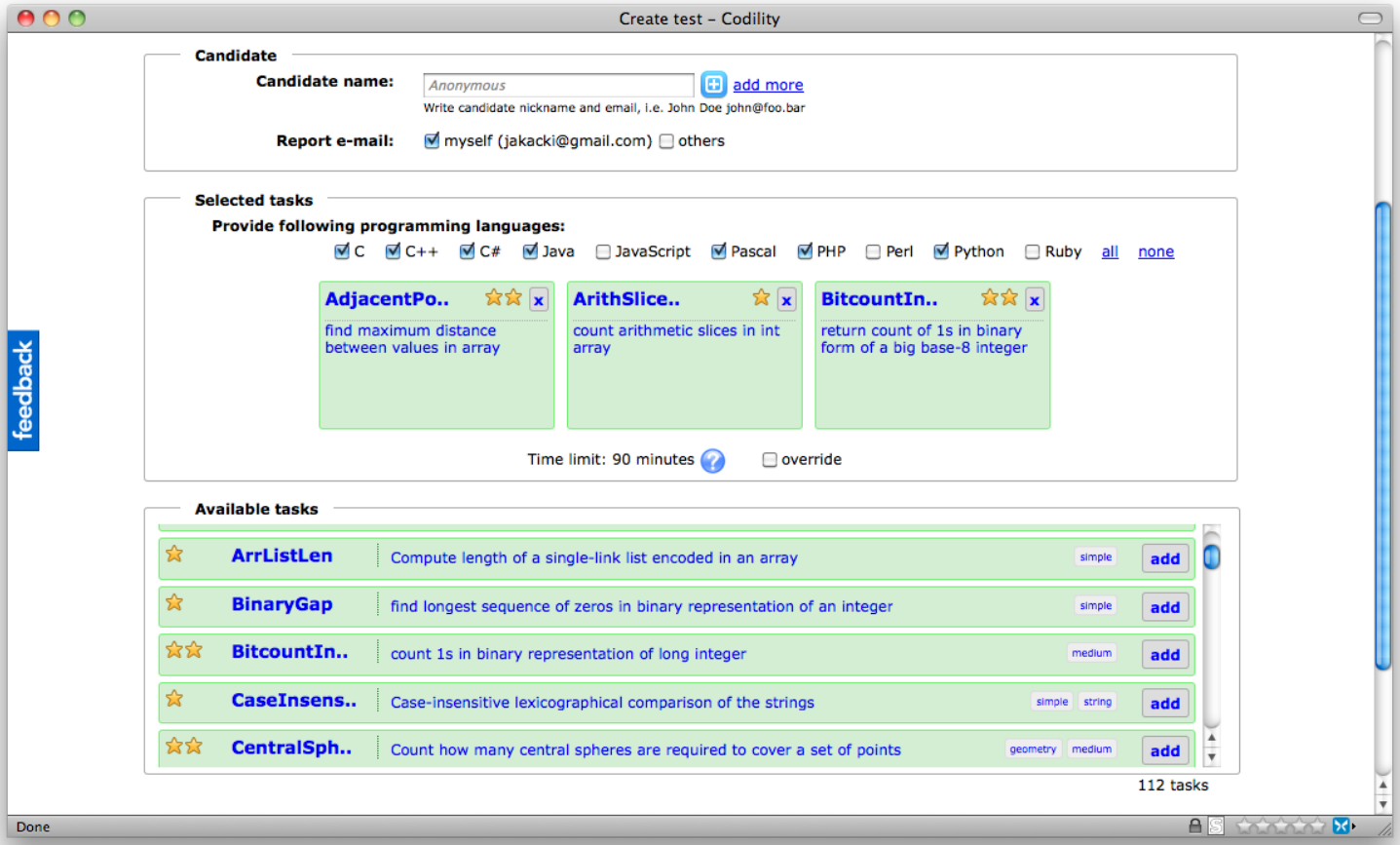
# Task Preparation for Multiple Programming Languages

Solution:

- Python — front-end programming language
- Automated translation of solutions written in (a subset of) Python into other programming languages
- Evaluation based on translated solutions in Python
  Language-specific solutions are also possible
- Automated time-limits calibration — independent of particular programming language and hardware

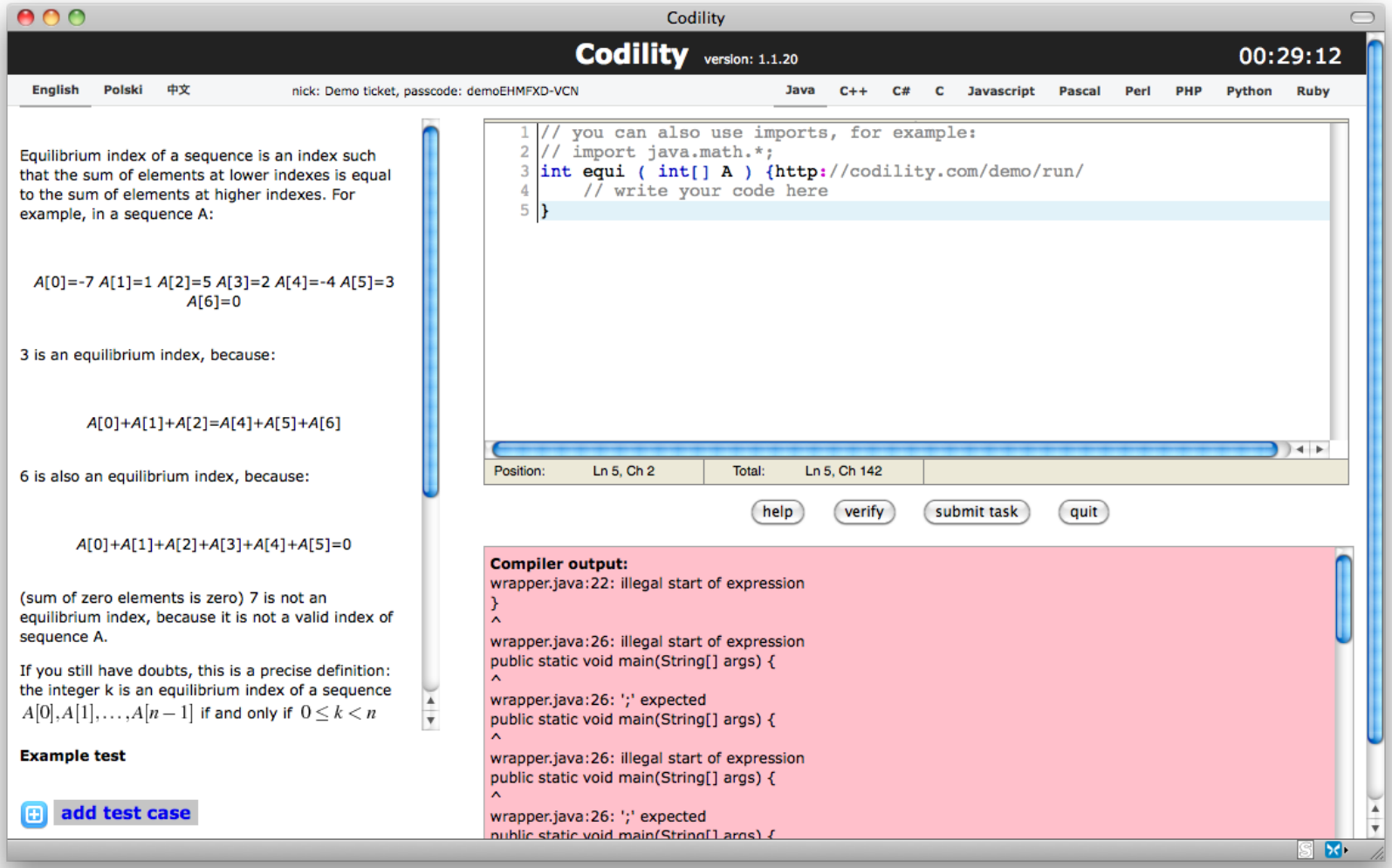
# Codility Web Service

- Two types of users: candidates and recruiters
- Recruiter's interface:



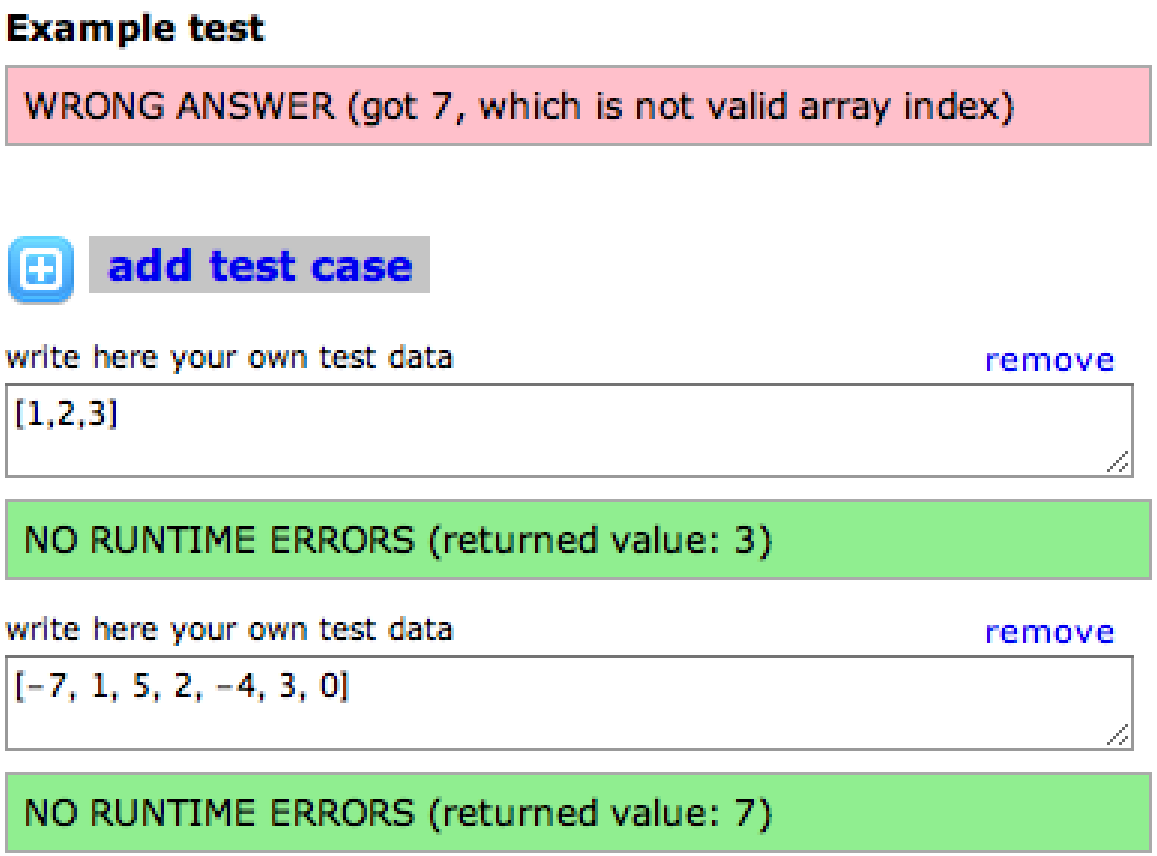
# Codility Web Service

- Candidate's interface — programmer's micro IDE:



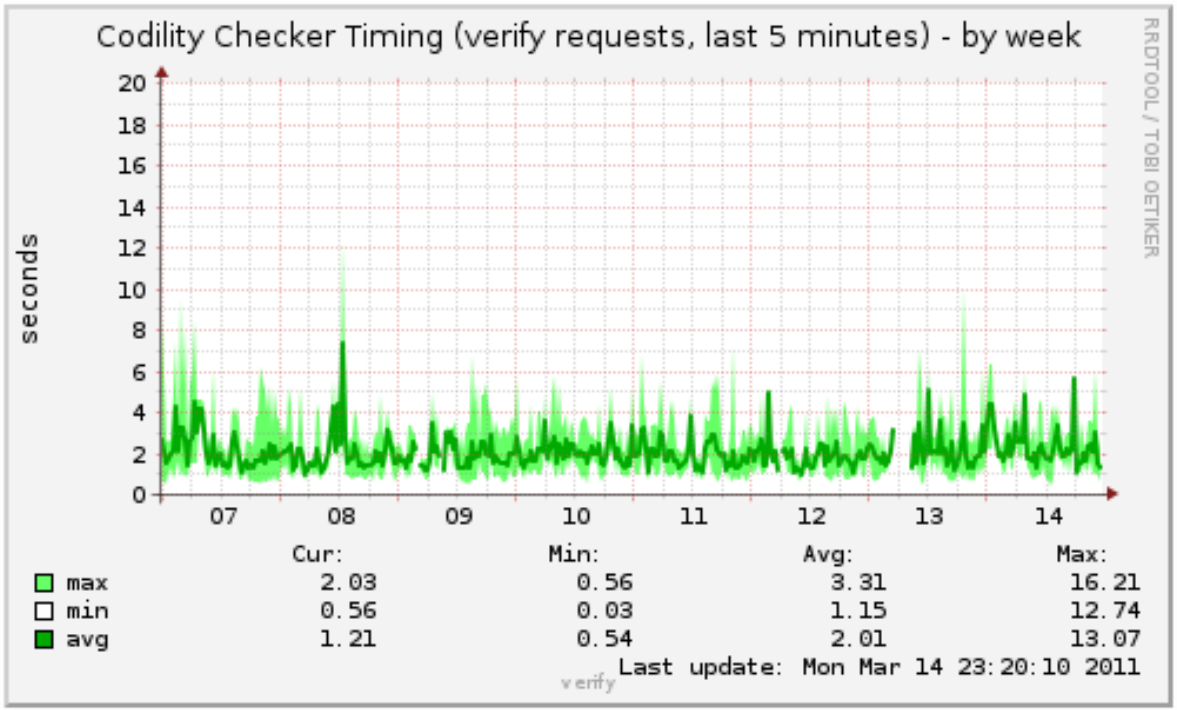
# Codility Web Service

- Candidate's interface — testing facility:



# Evaluation Technology

- Generally, based on IOI and ACM CPC models
- Experimental asymptotic time complexity assessment
- Reactive tests augmentation — hybrid of:
  model checking, symbolic interpretation, random and enumerative test generation
- Scalability — real time evaluation
- Grading machines — Amazon EC2 computing cloud



# Correctness and Soundness

- Goal: to distinguish good programmers (winners) and bad programmers (losers).
- What does it mean?
- **Correctness**: all winners are good programmers
  all bad programmers are losers
- **Soundness**: all good programmers are winners
  all losers are poor programmers
- Programming contests — focus on correctness,
  occasional lack of soundness is acceptable
- Pre-hire screening — soundness $>$ correctness
  Codility's correctness $> 90\,\%$
- 50 candidates $\rightarrow$ 5 winners $\rightarrow$ 1 hire

# Scoring

Scoring:

- Similar to IOI model
- 1–6 problems to solve
- Total score is the sum of scores for the tasks

# Conclusions

- Successful application of programming contest's technology to pre-hire screening
- Today Codility is a profitable business
- Codility's customers include:
  Nokia, Siemens, Barnes & Noble
- Two innovative technologies:
  - automated time complexity assessment, and
  - reactive tests augmentation

# Certify Yourself

**Want to try it yourself?**
**Get certified at:**
`http://codility.com/cert/start/`

Thank you for your attention!