



Competitive programming and the ACM-ICPC contest

Competitive programming

- Solve a set of problems in a limited amount of time.
- For each problem, write program that given any input (within certain limits), outputs the correct solution, while also respecting constraints on memory and running time.
- Usual solutions can range from 10 to 300 lines of code.
- Receive sample input and output.
- Actual evaluation is done on hidden inputs.

Competitive programming

- The feedback and scoring differ from contest to contest
- Partial scoring (IEEEExtreme)
- Accepted / rejected scoring: you need to pass all tests (most contests)
- Live feedback (ACM - ICPC)
- Partial feedback from small number of hidden tests (Codeforces)
- Completely hidden feedback (Topcoder)

The ACM International Collegiate Programming Contest

- Team contest: 3 coders, one keyboard
- 3 phases:
 - Sub-regionals
 - not that official, mostly used for training and for selecting teams
 - Regionals
 - NWERC (Northwestern Europe Regional Contest)
 - usually 1 or 2 teams per university
 - World Finals
 - the first few teams from each regional (somewhere between 1 and 10)



The ACM International Collegiate Programming Contest

- 7 - 14 algorithmics problems
- It's about thinking not about memorizing (you can bring 20 pages of reference material that you can fill with algorithms and formulas)
- Feedback almost instant:
 - Accepted (Yaaay!)
 - Wrong Answer, Time / Memory Limit Exceeded, Compilation / Runtime Error
- Languages: C, C++, Java (even if there were more, they would probably be too slow)
- Live scoreboard (except for the last hour so there's at least some surprise)

Problems

- You have N strings, count how many unique strings there are. ([link](#))
- You have N (≤ 10) pairs of meals, each having a sourness and bitterness value. Select any number of meals (> 1), such that the difference between overall sourness and bitterness is as close to 0 as possible. ([link](#))
- A list of bytes were scrambled by the operation $(x \ll 1) \oplus x$ aka $x \text{ XOR } 2x$. Given the scrambled bytes, recover the original ones. (J from [link](#))
- Given a log of messages and users who sent them, find the list of words that were used by all the users. (C from [link](#))

Why do competitive programming?

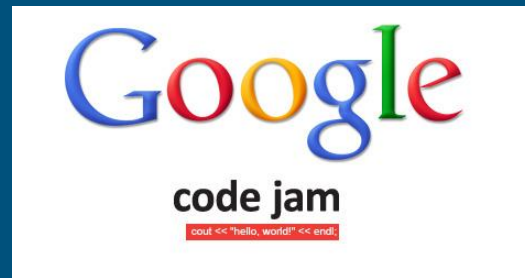
- Better problem solving
- Improve thinking and strategizing skills
- Coding - both writing and debugging
- Interviews
- Get prizes!
- It's fun and fulfilling!

Resources

- Luckily, there are lots of websites that provide and evaluate problems
- Codeforces:
 - hosts online competitions every few days
 - users get ratings
 - archive of over 300 previous contests, with the possibility of “virtual participation”
 - also hosts team competitions at their [gym](#)
- Topcoder
 - also hosts online competitions and rates users
 - archive of over 600 previous contests
- Other archives
 - [ICPC-archive](#)
 - [Kattis](#)
 - [Spoj](#)

Other contests

- Facebook Hacker Cup
- Google Code Jam
- IEEExtreme
- Bloomberg Codecon



Future meetings

- Next meeting?
- Who?
- How often?
- Where?
- Homework?



Questions?

