ACM International Collegiate Programming Contest

C. Maria Keet

Department of Computer Science, University of Cape Town, South Africa
mkeet@cs.uct.ac.za

Introduction, July 29, 2016

Outline

1. ICPC
2. Solving the problems
3. To do

Outline

1. ICPC
2. Solving the problems
3. To do

What is the ACM ICPC?

- Team competition: 3 students from the same university
- Usually 8 programming problems at Regionals and 12 in the Finals (examples later)
- 5 hours to solve as many of them as you can!
- Problems can be solved in Java or C/C++ (and Python sort of)

Competition structure

- Southern African regional
  - Around 60 university teams from SA
  - About 30 teams from rest of Sub-Saharan Africa
  - Compete at local sites (UCT is the local site for the Western Cape)
  - On a Saturday in late Sept/early Oct
- World Finals
  - About 125 teams from round the world
  - On site, in The USA in June 2017
  - Winning team from regional goes (if a UCT team wins, UCT will pay for your trip)

Competition structure

- Southern African regional
    - Around 60 university teams from SA
    - About 30 teams from rest of Sub-Saharan Africa
    - Compete at local sites (UCT is the local site for the Western Cape)
    - On a Saturday in late Sept/early Oct
- World Finals
    - About 125 teams from round the world
    - On site, in The USA in June 2017
    - Winning team from regional goes (if a UCT team wins, UCT will pay for your trip)

myicpc icpcnews.com
my ICPC
acm International Collegiate Programming Contest
IBM event sponsor
ДИВС
Tokyo Institute of Technology
Tsinghua University

Welcome
PARTICIPANT
2016 World Fin
Phuket
acm International Programmin
IBM

Why participate?

- Having fun solving puzzles!
- Learn about problem solving and programming along the way
- The challenge
- Great on CV if you do well
- Overseas trip if you win

Eligibility

- In short: “undergraduate”, meaning in SA system Bachelors and Honours students, up to <5 years full time
- Basic requirements, with, a.o.: Willingness to go to the Finals if you win the regionals (More detailed description: `http://icpc.baylor.edu/regionals/rules#HTeamComposition`)
- Eligibility period:
  - Began post-secondary studies in 2012 or later is eligible to compete,
  - Was born in 1993 or later is eligible to compete.
  - Can be extended, but has to be motivated (e.g.: illness, work/studies, or personal reasons); let me know asap if this applies to you

Eligibility

- In short: “undergraduate”, meaning in SA system Bachelors and Honours students, up to <5 years full time
- Basic requirements, with, a.o.: Willingness to go to the Finals if you win the regionals (More detailed description: `http://icpc.baylor.edu/regionals/rules#HTeamComposition`)
- Eligibility period:
    - Began post-secondary studies in 2012 or later is eligible to compete,
    - Was born in 1993 or later is eligible to compete.
    - Can be extended, but has to be motivated (e.g.: illness, work/studies, or personal reasons); let me know asap if this applies to you

Outline

1 ICPC

2 Solving the problems

3 To do

Problems (1/2)

- They are diverse: some require a bit of maths, many are algorithmic in nature
- Real problem to solve is always ‘dressed up’ in a little story; e.g.:
  - Sorting luggage going to different destinations
  - Making a beautiful necklace with coloured beads in symmetry
  - Wires crossing on a circuit
  - John and Mary with some fitness programme
  - Blowing up asteroids
  - Cutting Emmenthal cheese in a food processing plant

Problems (2/2)

- In some cases, a clearer task is described in such a text, e.g.:
  - Find the largest prime in a list of numbers up to 2^{32}
  - Find the first digit of B^N, given $1 \leq B \leq 10$ and $1 < N < 10000$
- In other case, underlying problem to solve, a.o.:
  - Graph traversal (shortest path [e.g., Dijkstra], spanning tree [e.g., Kruskall])
  - Sorting (heap sort, merge sort, etc.)
  - String matching/manipulation (e.g., Levenshtein distance)

Problems (2/2)

- In some cases, a clearer task is described in such a text, e.g.:
  - Find the largest prime in a list of numbers up to 2^{32}
  - Find the first digit of B^N, given $1 \leq B \leq 10$ and $1 < N < 10000$
- In other case, underlying problem to solve, a.o.:
  - Graph traversal (shortest path [e.g., Dijkstra], spanning tree [e.g., Kruskall])
  - Sorting (heap sort, merge sort, etc.)
  - String matching/manipulation (e.g., Levenshtein distance)

Scoring

- Submissions automatically marked
- Judges response is one of: Correct, Wrong answer, Format Error, Time-limit exceeded, Runtime error, Compile error
- Correct answer gets you a color-coded balloon
- Note: Just one test case wrong gets you an incorrect answer! you'll get a time penalty for that
- Teams ranked by number of problems solved
- Ties broken using “time penalty”

Teams

- Your team overall needs to have several skills: problem solving, designing, coding
- Only one computer per team!
- This leaves lots of room for interesting team make-up and team strategy
- Choosing which problems to solve first, splitting up the problems among the members, not all focusing on the same problem at once, etc requires a team strategy
- Being able to solve those problems, recognising basic algorithms and repurposing them for the task, and programming are skills that require practice

Teams

- Your team overall needs to have several skills: problem solving, designing, coding
- Only one computer per team!
- This leaves lots of room for interesting team make-up and team strategy
- Choosing which problems to solve first, splitting up the problems among the members, not all focusing on the same problem at once, etc requires a team strategy
- Being able to solve those problems, recognising basic algorithms and repurposing them for the task, and programming are skills that require practice

Outline

1. ICPC
2. Solving the problems
3. To do

Way forward

- There will be training days to help you prepare for the regionals (next 2 slides)
- Find team mates with some overlapping and some complementary skills
- Have a look at `http://acm.cs.uct.ac.za/`
- Chat with the previous contestants
- Practice: e.g., CodeForce, UVa Online Judge (and 1016Challenge for the 1st years)

What we'll cover in the training

- Problems in some detail
  - Problem-solving strategies in computing
  - Methodological approach to solving contest problems
  - Algorithms and data structures you may not have seen yet (e.g., for graphs, geometry, advanced string manipulation)
- The 'soft stuff'
  - How to work together in a team
  - Time management during a contest
  - Avoiding some common pitfalls of 'contest fever'
- Practice with practice contests

Scheduled training dates (tentative!)

- Aug 6: 10:00-16:00
- Aug 13: 10:00-16:00
- Aug 27: 10:00-16:00
- Sept 3 or 10: 10:00-16:00 (Ashraf Moolla, possibly)
- Sept 17: 10:00-16:00
- Oct 1: 10:00-16:00
- Oct 15: 10:00-16:00 (if the contest isn't held then)

* Changes will be communicated online and on the "contest" vula site (latter first)
** Training material will be put on the contest vula

Questions?