Solution Discussion / Basics

Wednesday, September 13, 2017 10:42 AM

Feedback
Did you have fun? What do you think we can organize better?
How many did you solve? How many did/do you have thoughts on? Did you read all the problems?
What did you spent/waste most time on?

Basics
Strategy: read all problems, test local, design test input, edge cases
IDE/Editor/Compiler: what's supported on regional, (esp. Python implementations)
Some information is here: https://icpc.baylor.edu/regionals/finder/north-central-north-america-2017. We'll have to ask them about certain specifics (e.g. Python implementation)
Techniques useful, I.e. input/output
Always use standard input and standard output for I/O
For massive inputs, optimized input may be useful
Printf/scanf are faster than cin/cout (C++)
BufferedReader/BufferedWriter are faster than Scanner (Java)

- Contest rules review
  - Write out example for penalty calculation on the board
- Complexity heuristics. Programs should run within roughly one billion operations

n	Worst AC algorithm
<1K	O(n^3)
<10K	O(n^2)
<100M	O(nlogn)
<1B	O(n), O(logn), O(1)

Input/Output tricks
STL for C++: vector, map, set, mulitiset, algorithm, etc

Solution Discussion, Explain each problem

- Fox
  - Input parsing
  - Many ways to solve, ask audience
- Weak vertices
  - Seems like graph, but you don't really need a graph
  - Check 3-clique
  - Talk about graph data structures
    - Adjacency matrix
    - Adjacency list, edge list, briefly
    - They already gave us adjacency matrix, so we should use that.
    - "Can I copy existing graph library code?" -- Contestant at tryout

        - Yes, in the real contest you can bring library code with you, but in most cases (this case included) it's too heavyweight to be worth using. Complex graph algorithms yes use library code, simple graph algorithms and graph data structures no.
        - We have a shared library.
    - Clarify sample input, the second case confused someone
- Dragon
    - (Strategy) Long description, easy solution
    - Greedy, ask audience
    - May need to sort for time limit
- 0-1 sequence
    - Combinatorics, ask audience for thoughts
    - Recurrence relation of the answer for prefixes. Jinman can explain
- Coalition
    - (Remark) Hard description
    - Brute force is too slow, O(2^n) where n<=150
    - Greedy? Ask audience
    - Knapsack
        - If you write it out as an optimization problem, it resembles Knapsack's formulation
        - Can be transformed into knapsack by inverting it and using log probabilities
- Erdös
    - Graph modeling + BFS, ask Yuxing
    - What if each line stands for authors for a paper, e.g. any two authors in each line are coauthors? Let me know if you have efficient solution.

Last three are harder ones. Make them brief if time is not enough.