In ACM ICPC

COMPETITIVE PROGRAMMING

Competitive Programming

- Given well-known Computer Science problems, solve them as fast as possible!
  - Not about “software engineering”
- Well-known = not research problems!
  - Problems in ACM ICPC have this characteristic!

Quick Test 1 – Algorithm Test

- Try to solve this problem as fast as possible
  - Given a connected graph, |V|=100K & |E|=|V|-1
  - Find 2 vertices a and b such that path between a and b is the **longest** among all other paths?
- Can you solve each it in under 1 hour?
  - Or 30 minutes?
  - Or perhaps… faster?

Quick Test 2 – Analysis (1)

- Which algorithm is the fastest?
  - Options for the sub-questions below
    - A. Use Sieve
    - B. Pre-calculate them
    - C. Use prime testing function
  - 1. Display first 10 prime numbers!
  - 2. Display first 10000 prime numbers!
  - 3. Display primes between large [A .. A+100]*!
  - 4. Check if a single number X is a prime!

Quick Test 2 – Analysis (2)

- Which data structure is better?
  - Given a list of lowercase character [‘a’..’z’] and character cost, e.g. ‘a’ → 5$, ‘b’ → 7$, …, ‘z’ → 3$. Output the cost of given sentence, e.g. “abz” = 15
    - A. Binary Search Tree
    - B. Direct Addressing Table
    - C. Hash Table
    - D. Array of Pairs <char, int>
    - E. Heap
    - F. Graph

Quick Test 2 – Analysis (3)

- Can this algorithm be used?
  - Given a Graph G of |V| = 50, check whether vertex i and j in G are connected using *Floyd Warshall* algorithm
  - Time limit = 1s, CPU speed is approximately 1 million operations/s
  - What if |V| = 100?
  - What if |V| = 10000?

Quick Test 2 – Analysis (4)

- What algorithm to use?
  - Given a sparse unweighted graph of $|V| = 10000$ and $|E| \leq 10*|V|$, what is the shortest path between vertex i and j in the graph?
    - A. Floyd Warshall
    - B. Breadth First Search (BFS)
    - C. Dijkstra
    - D. Depth First Search (DFS)

Question Types in ACM ICPC

- Ad Hoc
  - Straightforward
  - Simulation
- Complete Search
- Divide and Conquer
- Greedy
- Dynamic Programming
- Graph
- Mathematics-related
- String Processing
- Comp. Geometry

Programming Languages in ICPC

- You should master one (**preferably more**) programming languages
  - Reduce the amount of time looking at references
  - Use shortcuts, macros, avoid comments
  - Use libraries whenever possible
- Idea: once you figure out a solution for a problem, you are able to translate it into a bug-free code, and do it fast!

Quick Test 3 – Language (1)

- What is 777! (factorial of 777)?
  - 3! = 6
  - 4! = 24
  - 5! = 120, etc
- 25! = 15,511,210,043,330,985,984,000,000
  - Then, how about 777! ??
- Very long ……..
- How to solve this “problem” quickly?

Quick Test 3 – Language (2)

- Given *n* (*n* < 1 million), list *L* of unsorted *n* integers, and a set of values V, |V| < 10.000
- For each v in V, find *w* in *L* such that *w*+7 = *v*
  Output "-1" if we cannot find such *w*!
  - This is basically searching for *w* = *v*-7 in *L*
  - Linear scan can be slow although easy to code
  - Must use binary search, but not easy to code
  - Can use "sort" + "lower_bound" library in C++ STL

Judging System in ICPC

- Ultimately, we want "Accepted (AC)" result ☺
  - i.e. our code passes the judge's secret test data
- However, we may instead be given: ☹
  - Presentation Error (PE)
  - Wrong Answer (WA)
  - Time Limit Exceeded (TLE)
  - Memory Limit Exceeded (MLE)
  - Runtime Error (RTE)

Quick Test 4 – Challenge this Code

- Find a “bug” in this code. This is a “solution” for the question posed earlier. Hint: |L| = 1M

```
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
  int n, v, L[100], *ans;
  cin >> n >> v;
  for (int i = 0; i < n; i++)
    cin >> L[i];
  sort(L, L+n);
  cout << (((ans = lower_bound(L, L+n, v-7)) != L+n
    && *ans+7 == v) ? *ans : -1) << endl;
  return 0;
}
```

Tip: TopCoder has “Challenge” phase during Single Round Matches

Practice and More Practice

- Try them
  - University of Valladolid (UVa) Online Judge
    - http://uva.onlinejudge.org
    - http://acm.uva.es/archive/nuevoportal
  - PKU Judge, SPOJ, etc…
  - TopCoder
    - http://www.topcoder.com
  - USACO
    - http://train.usaco.org

CS3233 Text Book

- http://www.lulu.com/product/paperback/competitive-programming/12110025