Day 4 Editorial

April 29, 2016

ETH Zurich ACM ICPC Training Camp. April 2016

A. Barcode

Solution

- Dynamic programming
- Total complexity is $O(n)$

C. Capital

Solution

- That's just a tree dynamic programming exercise

```
void goUp(int v, int parent, int maxUp) {
  int max1 = -1;
  int max2 = -1;
  up[v] = maxUp;
  for (int to : edges[v]) {
          if (to == parent) continue;
          if (max1 < 0 || down[max1] < down[to]) {
                  max2 = max1;
                  max1 = to;
          } else if (max2 < 0 || down[max2] < down[to]) {
                  max2 = to;
          }
  }
  for (int to : edges[v]) {
          if (to == parent) continue;
          int maxDown = max1 == to ? max2 : max1;
          maxDown = maxDown >= 0 ? down[maxDown] : 0;
          goUp(to, v, Math.max(maxUp, maxDown));
  }
}
```

D. Piatra Neamt

Solution

- That's another tree dynamic programming exercise
- For every vertex compute `down[v]` — sum of distances going down in rooted tree
- And calculate `up[v]` — sum of distances that the one or more first edges on path going up

E. P = NP

Solution

- As you have to satisfy two of three
- Instead of `x`, `y`, `z` make
  1. `x OR y`
  2. `x OR z`
  3. `y OR z`
- Solve 2SAT problem

F. Connections

Solution

- Problem is just equivalent to number of correct bracket sequences
- Answer is catalan sequence
- Calculate using dynamic programming

G. Ivan's Game

Solution

- Cost of the move $(S_1 - K_1) \times (S_2 - K_2)$
  - Subtract 1 from all the values, so cost of the move is now $S_1 \times S_2$
- There's no profit making moves $K_1 > 1$ and $K_2 > 1$
  - Sum of all pairs of numbers added
  - Can divide each sequence into two, will be better
- If $K_1 = 1$ or $K_2 = 1$ it's just sum of multiplications of a number from one of the sequences to some numbers of other sequences
  - The same as taking pairs of numbers one by one
- So do dynamic programming `f[a][b]` — the smallest sum can make if all numbers $i > a$ in first sequence and $j > b$ in second sequence are taken
- You either take both, or one of the numbers

H. Hang

Solution

- State is (values of registers, instruction pointer). Total number of states is $2^{32} \times 16$ which is too much.
- Not all registers are important
- If we have `jz x a` command than `a` register is important
- If we have `MOV a b` command and `a` is important than `b` is also important
- Same rules for other instructions (one instrction add not more than one important register)
- Not more than 16 important registers
- Total number of interesting states is $2^{16} \times 16$. So we can do bfs on all interesting states.

J. Packing Trees

Solution

- The problem is to color tree into several colors, so that expected number of color changes on path from vertex to root is minimized
- First observation: every color makes connected set of vertices
  - If not, you can make a new color for one of the regions, answer won't change

J. Packing Trees

Solution

- You can calculate the expected number of times each edge is traversed
- Do the dynamic programming: `f[v][B]` — the expected number of color changes in subtree rooted at v, so that the number of vertices colored to the same color as root is B
- For every of children you either start new color component, then you added number of times you traversed the edge
- Or continue the same color
- Do the internal DP: `g[i][B]` — the expected number of color changes in the first i children so that B vertices of the same color used

K. PAM

Solution

- We need to solve a system of linear inequalities
- Common way of doing this is Simplex algorithm or Ellipsoid algorithm
- But this task also can be solved with approximate algorihtms

L. Robots on a Board

Solution

- If robot repeats his list of commands 2^{24} times, he will fall of the board
- So we need only $2^{24} \times 256$ moves to make
  - This will exceed the time limit
- Robots can collide in next 256 moves only if the distance between them is not more than 256
- Let's precalculate for every of second robot's position respective to first robot (256×256 positions) will they collide and when
  - This is calculated just simulating for all the starting respective positions
- Then we can simulate robots' programs in $O(1)$
  - Lookup will they collide in one iteration
  - If not change their position
  - Repeat 2^{24} times

M. Safe Cracking

Solution

- That is the standard problem
- The optimal final position of all holes coincide with one of the initial positions
- $O(n^2)$ solution is just try all of them, for every other compute minimum of two distances and sum up
- To do $O(n)$ you can use two-pointers technique, the elements form two segments in cyclic array, if you sort all the numbers initially
- Accurately implement

N. Seti

Solution

- The problem says that given some polynomial values in calculated modulo some prime in some points, get the polynomial
- `https://en.wikipedia.org/wiki/Lagrange_polynomial`
- You can divide modulo prime

O. Stairs

Solution

- The key idea is just when two guys meet, they keep moving not interfering with each other

P. Protect the Statues

Solution

- Problem is to find the area of convex hull of all circles
- Since we need only $\frac{1}{10}$ accuracy and coordinates are up to 10^4
- Get 1000 points on each circle
- Build convex hull
- Output area of the polygon

Q. Street Directions

Solution

- Create two edges for every bridge in both directions
- Take biconnected component, use DFS to direct the edges:
  - Build spanning tree using DFS
  - The edges in spanning tree direct from it's parent to child
  - The other edges direct from descendants to ancestors

R. Word Rings

Solution

- Build graph:
  - 26^2 vertices for all pairs of letters
  - Every word is an edge from its first two letters to its last two
- Problem is to find the minimum mean cost cycle
- Answer is: $\min\limits_{v} \max\limits_{i} \frac{d_{n,v}-d_{i,v}}{n-i}$
- Where $d_{i,v}$ — is the shortest path from some vertex s to v with i edges
- s is the vertex that everything is reachable from it
- For more information find papers about minimum mean cost cycle