East Central North America Regional 2006

Solution Sketches

Renzo Gonzalo Gómez Diaz

Disclaimer *This is an unofficial analysis of possible solutions to the problems of the East Central North America Regional 2006. Any error in this text it's my fault. If you find any error or know of other ways to solve this problems, I'd happy to know about it at* `gomez.renzo@gmail.com` *or my blog.*

A. Caterpillar

To check that the input graph is indeed a tree, we have to ensure that it's connected and acyclic. Then, note that every vertex that isn't a leaf must be on the spine of the caterpillar. So, for every non-leaf vertex, we check that the number of neighbors with degree greater than one is less than 3.

B. Hie with the Pie

Begin by calculating the shortest path between any pair of vertices (for example with the Floyd Warshall algorithm). Then, test every possible ordering of delivery, and output the minimum cost.

C. Mahershalalhashbaz, Nebuchadnezzar, and Billy Bob Benjamin Go to the Regionals

At first sight, it seems that a greedy strategy like the following would work

GREEDY: Sort the words by its lenght in non-decreasing order. After that, a group consist of k adjacent words beginning from the first one.

But, this is a counter-example for $n = 8$ and $k = 4$

$$\{1, 1, 4, 5, 5, 6, 6, 6\}$$

The greedy solution gives a negative answer for this test case. Since the mean of the group $\{1, 1, 4, 5\}$ is a number between 2 and 3, and 5 is too far from it. But, the grouping $\{1, 1, 5, 5\}$, $\{4, 6, 6, 6\}$ is a valid solution. Furthermore, the official solution to this problem also uses the greedy strategy mentioned before. So, it seems that the test cases on the online judge are weak. Any hint for a polynomial solution to this problem is welcome :).

D. The Mastermind Master's Mind

First, generate all possible words. Lets call that set W. Also, let S be the set of valid solutions to the game (given the initial guesses). Finally, test every possible $w \in W$, as a guess and calculate in how many classes does w partition S.

E. Plaque Pack

In this problem we have to simulate the way the plaques are stacked. No trick just simulate.

F. Roofing It

Let $P = \{p_1, \ldots, p_n\}$ be the set of points and denote by $U(P)$ the upper hull of P. Observe that, by the convexity constraint, the roof must be formed by a subset of the lines contained in $U(P)$. So, first we find the $U(P)$ using a convex hull algorithm (like the monotone chain algorithm).

Let $\mathcal{L}(i, j)$ be the set of lines belonging to the $U(P)$ that pass through two points of $\{p_i, \ldots, p_j\}$. After finding $U(P)$, calculate for every interval (i, j), $i \leqslant j$

$$C(i,j) \quad = \quad \min\{\, \max\{d(p_k, L) : i \leqslant k \leqslant j\} : L \in \mathcal{L}(i,j)\,\}.$$

where $d(p, L)$ denotes the vertical distance between point p and line L, and $C(i, j) = +\infty$ if $\mathcal{L}(i, j) = \varnothing$. Finally, lets define the following recurrence

$$dp(i,k) \quad = \quad \text{minimum distance of } \{p_i, \ldots, p_n\} \text{ using } k \text{ lines.}$$

Using $C(i, j)$, we can calculate the values of the recurrence using dynamic programming

$$dp(i,k) \quad = \quad \min\{C(i, j-1) + dp(j, k-1) : i < j < n\}$$

G. Snakes on a Plane

To begin with, find the connected components with any graph search algorithm and check which ones are valid snakes. For each valid snake (path) test if it can be extended by any of its ends.

H. Stake Your Claim

Let E be the set of empty squares. Then, number each element of E arbitrarily from 0 to $M - 1$. It's important to observe that there is a bijection between a game state and a M digit number in base 3. Let $D = d_{M-1}d_{M-2}\ldots d_0$ be a number in base 3, then it can be mapped to a game state in the following way

$$d_i = \begin{cases} 2, & \text{square } i \text{ is an empty square,} \\ 1, & \text{square } i \text{ is filled with } 1, \\ 0, & \text{square } i \text{ is filled with } 0. \end{cases}$$

Also, note that if two states have the same parity of empty squares, then it's the same player's turn. Let S and T be two different states. We'll say that T is a successor of S, if T

can be reach in one move from S. Lets denote by $N(S)$ the set of successor states of S, we define the following recurrence

$$dp(S) = \begin{cases} \max\{dp(T) : T \in N(S)\} & \text{it's player 0's turn,} \\ \min\{dp(T) : T \in N(S)\} & \text{otherwise.} \end{cases}$$

For the base case (state without empty squares), $dp(S) = m_0 - m_1$, where m_i is the area of the largest connected region filled with player's i number for $i = 0, 1$.