Day 5: Problem Analysis

10.05.2014

Problem A+. Consanguine Calculations

Problem A+. Consanguine Calculations

Problem statement

- You are given blood types for two of pair of parents and child.
- You need to find what could the blood type of the third person be

Solution

- Try all blood types
- Have to check whether two bloodtypes can generate third

Problem B. Containers

Problem B. Containers

Problem statement

- You are given a string of english letters
- You need to put letters in several stacks, so that letters in every stack are in sorted order
- You have to put letters in order that they are given in the string
- What is the minimal number of stacks do you need

Solution

- Gready solution works
- Put the next letter to stack that has the least number on its top
- Or if there is no such, just create new stack and put in it

Problem C. Grand Prix

Problem C. Grand Prix

Problem statement

- After parsing the statement the problem is following (Don't forget $\varphi = 0$ case)
- You are given several directed line segments on the plane
- You have to rotate the plane, so that for every segment $(x_1, y_1) \to (x_2, y_2)$, following is true $x_2 \geqslant x_1$

Solution

- For every segment find the set of angles, that can plane be rotated by, so that this segment is correct
- Every such set is $[\alpha - \frac{\pi}{2}, \alpha + \frac{\pi}{2}]$, where alpha is angle between the segment and X-axis
- Intersect all such sets
- Find the minimal angle and make answer

Problem D. Jacquard Circuits

Problem D. Jacquard Circuits

Problem statement

- You are given lattice polygon with no self-intersections
- You are to find how many lattice points are inside m smallest lattice polygons that are similar to given one

Solution

- Remove points that join two neighbouring and collinear sides
- Then find the smallest similar lattice polygon, by computing GCD of all the coordinates of vectors that form the given polygon
- Use Pick's theorem to find the number of lattice points inside polygon: $A = i + \frac{b}{2} - 1$, where A is area and b is number of lattice points on the border
- Let A_1 and b_1 be the numbers above for smallest polygon, then $A_v = v^2 A_1$ and $b_v = v b_1$
- Calculate the answer for every of m polygons using the formula

Problem E. Collecting Luggage

Problem E. Collecting Luggage

Problem statement

- You are given a polygon p, point on its border — q and point outside of the polygon — r
- Point q moves around polygon in counterclockwise direction at speed V_q
- Point r can move outside of the polygon at speed $V_r > V_q$
- What is the minimal time point r needs, to reach point q

Solution

- Suppose point r can make it in T seconds
- Then for all $x > 0$ it can make it in $T + x$ seconds, too, because $V_r > V_q$
- So make binary search for value T
- Now need to check, can it be reached it T seconds
- We know the position of point r in T seconds
- Build a graph, where vertices set consist of vertices of the p, point q and new position of point r
- Find shortest path and check if it can be done in T seconds

Solution

- How to build the graph?
- For every pair of vertices you have to check, whether point r can move along this segment
- Take every segment, that doesn't contain internal point that is on the border of p and any of internal points lie outside p
- Every path in the graph is formed by some subset of these segments
- Just have to check intersection of segments and if the point lies inside p

Problem F. Containers

Problem F. Containers

Problem statement

- You are given a game, game field is $n \times n$ and it contains m pairs of marbles and holes ($n \leqslant 4$)
- Game field also contains walls, and is surrounded by a wall
- You can lift a side of the game board and all the marbles will roll towards the opposite side until it reaches the wall or other marble
- You have to find minimal number of moves you need so that all marbles will make it to their holes

Solution

- Every state of the game can be represented as the position for every marble on the field
- Just use BFS algorithm on the graph, where vertices are states of the game and edges are the moves

Problem G. Network

Problem G. Network

Problem statement

- There are not more than 5 messages
- Each message is splitted into packets
- You have a buffer, that can store packets
- Output channel must accept packets of one message continuously and in order of its bytes
- As packets come to you in order you have to put them to the buffer sometimes
- Minimize the buffer size

Solution

- Try all permutations of messages you have
- Pass every message greadily, once you can pass packet of this message — do it, otherwise but input packet to the buffer
- Gen minimum buffer size over all permutations

Problem H. Raising the Roof

Problem H. Raising the Roof

Problem statement

- You are given triangles in 3D, their interiors don't intersect
- You are to find the are of the visible part of these triangles

Solution

- Let's solve for every triangle t_0 independently
- First let's find the projections of everything to OXY plane, find are of projection, then just divide by cosine of angle between the plane that this triangle is into and OXY
- Let's get all triangles s_i, that t_0 intersects (from now on we work in 2D) with and s_i is above t_0 in 3D meaning
- We need to find any point of intersection of two triangles and check z-coordinate for any interior point in intersection
- Then let's take all the vertices of s_i and t_0, and all points of intersections of their sides
- And taking all triangles' edges as graph edges, we get planar graph

Solution

- Every face of this planar graph is covered by some subset of triangles
- We need to find the area of faces that are covered by t_0 and only by it
- Well, how to get faces?
- Make two directed copies of every edge
- Just get any vertex, and any edge from it
- Next go only left every time, until we come to the first vertex, removing all edges visited.
- Face is possibly not convex polygon
- We have to find if there is an interior of intersection of not convex polygon and triangle

Solution

- How to intersect?
- Triangle and not convex polygon do intersect if:
  - Check if there is a point of not convex strictly inside the triangle
  - Check if any internal point of triangle is inside the polygon
  - Check if two sides have internal intersection

Problem J. Tunnels

Problem J. Tunnels

Problem statement

- You have undirected graph
- Spy is in vertex 1
- You can remove any number of edges any time
- What is the minimal number of edges you have to remove at worst case, so that spy won't be able to get to vertex 0

Solution

- If you could remove edges only before spy starts moving, then the problem is just in finding minimal $\langle S, T\rangle$-cut
- Idea: there is always optimal answer, that you have to remove all edges that you remove at once
- Suppose that is true, then the algorithm is the following
- $f(v)$ — is the answer to the problem if spy starts in vertex v
- $c(v)$ — is minimal $\langle S, T\rangle$-cut, for v and 0 being s and t respectively
- $f(v) = \max\limits_{uv \in E} \min\{f(u), c(v)\}$, you either cut it in v, or spy goes to the vertex so that the answer is maximal possible
- Use Dijkstra-like algorithm to find this value

Solution

- Let's prove the idea
- Suppose you have to remove edges more than once, and suppose that you do it not more than k times
- Suppose that for every $k < i$ found $f_k(v)$ are all correct
- Let's prove that for $k = i$ $f_i(v)$ is also correct
- Let u be the vertex the spy in, that we remove edges for the first time
- Suppose we removed q edges, then every $c(v)$ decreased by no more than q, so all $f(q)$ decreased by no more than q, so $f_i(v) = f_{i-1}(v)$