Day 2 Editorial

April 27, 2016

ETH Zurich ACM ICPC Training Camp. April 2016

A. Bidding

Problem statement

- Find the bid using the rules described in the statement

Solution

- Solution — carefully implement rules described in the statement
  - Calculate hcp
  - Calculate distribution
  - Implement all 11 rules

B. Core Wars

Solution

- Since "Each simulation continues running until either warrior executes a DAT instruction or until a total of 32000 instructions (counting both warriors) are executed" one can simulate the process
- Solution — carefully implement MARS

C. Doors and Penguins

Problem statement

- You are given two sets of rectangles
  - Sides parallel to coordinate axes
- You are to check if these sets can be separated from each other by a straight line
  - Line should not touch any of rectangles

Solution

- Find convex hulls of both sets
  - Replace each rectangle by its 4 vertices
- Check if convex hulls have a common point
  - Use linear Minkowski sum algorithm to check whether convex polygons have common point

D. Falling ice

Problem Statement

- Given a sequence of circular stones falling into the rectangular area
- We always put a stone in the downmost place we can, if there are several, the leftmost of those is chosen
- Output the coordinates of the centers of circles

D. Falling ice

Solution

- Let's simulate the process
- We already put some of the stones, put one more
- Consider r is the radius of new circle
- What are the points, which can contain the new circle
  - Expand all circles by r and shift all borders by r
  - All the points, that are inside the region and outside of expanded circles are valid
    - Intersect all pairs of objects: circles and borderlines, one of the intersections will be with minimal y-coordinate
    - One can see, that new circle always touches at least two objects

D. Falling ice

Can we move the stone to that point?

- Let's build a graph: vertices — all intersection points
  - Edges are build as below:
    - Get all points lying on one of the objects: either circle or line
    - Sort all the points: by angle or just by coordinate, if it's line
    - Get all neighbouring points and check, whether the arc or a line segment between them are in valid region (just check, if it's not contained in a circle and it is contained in bounded region)
    - You can get only one middle point of arc or line segment to check
- If the point is in the same connected component as highest valid point, then you can reach it

E. Human Knot

Problem statement

- $n \leqslant 500$ vertices are located on the circle
- Each vertex is connected to 2 other vertices
- You can move vertex to some other position on circle
- You want to obtain n-sided polygon without intersecting edges in smallest possible moves

E. Human Knot

Problem statement

- If there are more than one component — no solution
- Otherwise let's fix some set of vertices which will not be moved
- It should be already sorted if we count from some vertex
- You need $n - sizeofset$ moves to reorder other vertices
- So you want to find largest such set of vertices
- Try to split array of vertices indexes in all positions (we need to do this because array is actually cycled) and find longest increasing (decreasing) subsequence

F. Knots

Problem statement

- There is a process to make a graph of n vertices
  - n is even
- Two perfect matchings are chosen in a full graph with n vertices uniformly and independently
- Find the probability of the union of these perfect matchings to form a single cycle

F. Knots

Solution

- The number of ways to choose perfect matching is: $(n-1)\cdot(n-3)\cdot\ldots\cdot 3$
  - There are $n-1$ ways to choose pair for vertex 1
  - There are $n-3$ ways to choose pair for vertex with smallest unmatched vertex, and so on..
- Suppose one of the perfect matchings is chosen, every first matching gives the same probability
- The number of second perfect matchings that satisfy the first is: $(n-2)\cdot(n-4)\cdot\ldots\cdot 2$
  - There are $n-2$ ways to choose a new pair p_1 for vertex 1
  - You can't add an edge to form a cycle
  - There are $n-4$ ways to choose a new pair for vertex, which is the pair of p_1, and so on..
- Answer is $\frac{(n-1)\cdot(n-3)\cdot\ldots\cdot 3}{(n-2)\cdot(n-4)\cdot\ldots\cdot 2}$

G. Marbles in Three Baskets

Problem statement

- You have three baskets with identical marbles
- You can move some marbles from one basket to another, so that this doubles the number of marbles in a target basket
- Find the minimum number of moves to make all three baskets containing equal number of marbles

Solution

- Use breadth-first search on graph of states
  - Vertices — the triples of numbers, how many marbles each basket contains
  - Edges — moves

H. Margaritas on the River Walk

Problem statement

- You are give $n \leqslant 30$ objects with costs upto 1000
- You have $D \leqslant 1000$ dollars
- Find number of sets of objects with total cost $\leqslant D$ and you can't add one more object in a way that total cost is still $\leqslant D$

H. Margaritas on the River Walk

Solution

- Sort objects in decreasing order of their cost
- Add objects to knapsack in this order
- Before processing object i let's count all sets where i — cheapest not used object
- So we use all objects which are cheaper than i-th and some objects calculated with knapsack
- Total cost should be in range $(D - c_i..D]$
- Don't forget about case when you can buy everything!

I. String equations

Problem statement

- You are given $n \leqslant 100$ string of length $\leqslant 10$
- Not more than 10 distinct characters are used in strings
- You need to put each string on left or right side of equation zero or more times
- For each character total number of occurences on left and right side should be equal

I. String equations

Solution

- Each string is an array of 26 integers
- You need to find coefficients before vectors such that its sum equal to zero vector
- Atleast one coefficient should be non zero
- Do Gaussian elimination in long longs to find coefficients

J. Yellow Code

Problem statement

- You need to find a permutation of all n-bit words
- There are atleast $\lfloor n \rfloor$ different bits in a_i and $a_{(i+1) \bmod s}$ for all i in $[0..s)$, where $s = 2^n$

J. Yellow Code

Solution

- Let's solve it by hand for $n \leqslant 3$:
- 0, 1
- 00, 01, 10, 11
- 000, 011, 101, 110, 001, 010, 100, 111

J. Yellow Code

Solution

Get code for $n + 2$ bits from code for n bits:

00 01 11 10 ..	Code(n)
01 11 10 00 ..	Code(n)
11 10 00 01 ..	Code(n)
10 00 01 11 ..	Code(n)

K. Toothpick Arithmetic

Problem statement

- You are given an interger $n \leqslant 5000$
- You need to represent it as a sum of several terms
- Each term is a product of several numbers
- Sum of numbers should be minimized

Solution

- Use dynamic programming
- $product[i] = \min(i, \min\limits_{\substack{j<i \\ i \bmod j = 0}} (product[j] + product[i/j] + 2))$
- $sum[i] = \min(product[i], \min\limits_{j<i}(product[j] + sum[i-j] + 2))$
- $sum[n]$ — answer