Andrew Stankevich Contest 45

Problem analysis

Artem Vasilev   Pavel Krotkov

Peking University Camp, April 2016

A. Analogous Sets

Problem statement

- Two sets A and B of size n are called *analogous*, if multisets $A + A$ and $B + B$ are the same.
- $A + A = \{x + y \mid x, y \in A, x \neq y\}$

A. Analogous Sets
Solution

- No solution when n is not a power of 2.
- Solution for $n = 2$ is given in the sample test.
- Let A_k and B_k be the solution for $n = 2^k$. Set $A_{k+1} = A_k \cup \{x + m \mid x \in B_k\}$ and $B_{k+1} = B_k \cup \{x + m \mid x \in A_k\}$ where $m = \max A_k \cup B_k = 2^{k+1}$.
- Assuming there is a bijection between $A_k + A_k$ and $B_k + B_k$, it is possible to construct a bijection between $A_{k+1} + A_{k+1}$ and $B_{k+1} + B_{k+1}$.

B. Bayes' Law

Problem statement

- Given a random variable, find a segment $[L, R]$, such that $P(L \le x \le R | a \le f(x) \le b)$ is at least α.
- Random variable is a piecewise linear function of x $(0 \le x \le X)$.

- Conditional probability formula:
  $P(L \le x \le R | a \le f(x) \le b) = \frac{P(L \le x \le R, a \le f(x) \le b)}{P(a \le f(x) \le b)}$
- Set $x \mid a \le f(x) \le b$ can be represented as a union of at most n segments.
- $P(a \le f(x) \le b)$ is equal to total length of those segments and does not depend on L and R.
- Problem can be reformulated as follows: given n segments on line, find the shortest segment $[L, R]$ such that length of intersection of $[L, R]$ and these n segments is at least C.

B. Bayes' Law

Solution

- Proposition: there is an optimal answer $[L, R]$, such that L or R coincides with the beginning or ending of some segment.
- Consider segment $[L', R']$ and try moving it left or right while the length of intersection is not decreasing.
- Try all possible points for L and find the minimum R that achieves the required intersection length. Do the same for R.

- Count the number of sequences of length n with some properties.
- **Not** the Catalan numbers.

C. Catalian Sequences

Solution

- Use dynamic programming to calculate the answer.
- Need to define the *state* of DP, which should be less than whole sequence.
- What we are interested in:
  - Length of the current sequence
  - Number of ascends
  - Last element of the sequence
  - Minimum possible next element (maximum of all elements a_i where exist $j > i$ and $a_j > a_i$)
  - Set of all elements for which there's no such element
- This set of properties is enough to make a transition.
- Use BFS to only calculate reachable states or precalculate all the answers offline.

D. Drunkard

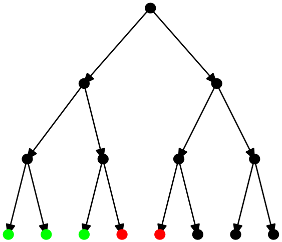
Problem statement

- Build a directed graph
- Two terminal vertices
- Two edges from all vertices except terminal
- Walker walks randomly
- Probability to end up in one of terminals $\frac{p}{q}$
- $p, q \leq 100$

D. Drunkard

Problem solution

- p green vertices
- $q - p$ red vertices
- Green leafs lead to the home
- Red leafs lead to the bar
- Black leafs lead to the root
- $V \leq (p + q) \times 4$



E. Elegant Scheduling

Problem statement

- We have an array of jobs
- We can switch first half with second, first quarter with second, third quarter with fourth, etc.
- Each consequent pair of jobs in final array costs $c_{i,j}$
- We need to minimize total cost
- $n \leq 4000$

E. Elegant Scheduling

Problem solution

- Dynamic programming
- $f_{i,j}$ — minimum possible cost of first i jobs if the number j is located at position i
- For every pair of number and position we have some set of numbers which can stay at previous position
- Let's just check values of f_{i-1} for all of them
- Total size of such sets is $n \times log_2 n$
- Total complexity — $O(n^2 \times log_2 n)$

F. Flights

Problem statement

- Undirected graph ($V \leq 1000$, $E \leq 100000$)
- Every vertex has is connected with first vertex
- Assign numbers to all edges
- For every pair of vertices sum of numbers on incident edges should be different

F. Flights

Problem solution

- Assign numbers from 1 to $E - V + 1$ to all edges except ones from first vertice
- Calculate sum of numbers of incident edges for every vertice
- Sort all vertices except first by this sum
- Assign the rest of the numbers in corresponding order
- Any vertices with equal sum will have different sum
- Any vertices with different sum will have even more different sum

G. Genome of English Literature

Problem statement

- We had some reasonable English text (50000 characters)
- We have 20000 randomly chosen pieces of length 50
- We need to build 100 pieces of length 500
- We need to cover at least half of the text

G. Genome of English Literature

Problem solution

- Let's say we have some substring of t already
- We want to increase it to the right
- We need to find a piece of length 50, which prefix of reasonable length is the same as the suffix of our substring
- We definitely have such piece (for more then $\frac{1}{3}$ of all positions in t there is a piece starting in this position)
- Can perform the search with hash tables

H. Hide-and-Seek

Problem statement

- Given a polyline, choose a maximum amount of corners, so no two of them see each other.

H. Hide-and-Seek

Solution

- For each pair of points determine, if they see each other.
- Run dynamic programming: $dp_{l,r}$ is the maximum number of corners we can choose from points with indices from l to r.
- $dp_{l,l}$ is always 1.
- If points l and r see each other, then both of them can't be included in the answer. So the optimal answer can be achieved by throwing one of them away: $dp_{l,r} = \max(dp_{l+1,r}, dp_{l,r-1})$

H. Hide-and-Seek

Solution

- Otherwise, they don't see each other. Let m be the smallest index $l < m < r$ such that m and r see each other.
- We state that every point from $[l, m-1]$ can't see any point from $[m+1, r]$, so these two segments are independent. Therefore, we can relax the value as $dpl, r = \max(dp_{l,r}, dp_{l,m-1} + dpm + 1, r)$.

J. Japanese Origami

Problem statement

- We have a strip of paper
- We can fold it by some rules
- We need to get special pattern of mountain and valley creases

J. Japanese

Key idea

- We have three actions:
  - fold the most left crease (if $l_0 < l_1$)
  - fold the most right crease (if $l_{n-2} > l_{n-1}$)
  - fold two consequent creases (if $l_{i-1} \geq l_i \leq l_{i+1}$ and these creases are different)
- Any solution can start with one of these three actions
- Any of these actions give us the problem of smaller size

J. Japanese

Solution

- Try to do one of these three actions while we can
- If we can't and we are not done — the answer is `NO`
- Otherwise we found an answer

K. Kaballah for Two

Problem statement

- Convex polygon
- We need to fit 2 circles in there
- Circles must not override
- Circles must be of maximal radius

K. Kaballah for Two

Problem solution

- Binary search for an answer
- To check some answer r:
  - Shift all polygon sides on r inside the polygon
  - Find two most distant points
  - If distance between them is more then $2 \times r$ the answer fits