Day 1 Contest: 2020 Seoul Regionals

Moscow International Workshop 2020

November 28, 2020

A. Apartment Complex Problem

Keywords: trees, DP/LCA.

Let S be the set of vertices with apartments. If $a, b \in S$, and a vertex v lies on an the ab-path, then for any other vertex u either $dist(v, a) < dist(u, a)$ or $dist(v, b) < dist(u, b)$, thus v is a good place.
If there are no such a, b, then all of S lies in a single connected component T after erasing v. In this case, the neighbour u of v in T is closer to each vertex of S.
The answer is the number of vertices v such that after removing v at least two subtrees contain elements of S. We can find them by rooting the tree at an arbitrary vertex, and counting the number of elements of S in each subtree with DP.
Equivalently, the answer is the size of the smallest connected subgraph containing all of S.

Complexity: $O(n)$. $O(n \log n)$ is also fine.

B. Biofuel

Keywords: DP/shortest path

We should charge at any intermediate point we visit. Indeed, if we don't charge, then skipping this point is better by triangle inequality.
The key observation is: suppose that u follows v in an optimal path. Then the amount of energy we have when leaving v is either W or $dist(u, v)$.
Indeed, if u is the finish point, any excess charging can be skipped so that we finish with zero energy.
If $c_u \leqslant c_v$, then there's no need to charge over $dist(v, u)$ before going to u, since we can charge that amount at u instead. If we arrive at v with more than $dist(v, u)$ charge, we can instead just skip v.
If $c_u \geqslant c_v$, then any charge made in u can be instead made in v, unless we max out the capacity (then we leave v with W), or we no longer charge at u (then we can skip u).

Now, we only need $O(\Delta n^2)$ states (v, δ, E), where v is the current vertex, δ is the number of visited vertices, $E = dist(v, u)$ for some u, or $E = w$ (call these values *possible capacities* at v).
The forward transitions are:

- If $E = dist(v, u)$, go to $(u, \delta + 1, 0)$, and charge to the smallest possible capacity at u.
- If $E < W$, charge to the next possible capacity.
- If $E = W$, try to go to all u, and charge at u to the smallest possible capacity larger than $W - dist(u, v)$.

There is a total of $O(\Delta n^2)$ transitions. The optimal path can be found with DP/Dijkstra in **complexity** $O(\Delta n^2 + n^2 \log n)$.

C. Comparison Issues

Keywords: DP/patterns.

Only matches between i and $i + 1$ are not predetermined, we can ignore all other matches.
Let d'_i be $r_1(i) - r_2(i)$ (without taking the absolute value) ignoring matches with $i + 1$. Let $dp_{i,k}$ be 1 if there is an arrangement of matches such that $|r_1(j) - r_2(j)| = d_j$ for all $j < i$, and $d'_i = k$.
$dp_{i,k}$ can be computed with DP. Initially only $dp_{1,0} = 1$.
To go forward from $dp_{i,k} = 1$, try all outcomes $s_1, s_2 in \{0, 1\}$ of the two matches between i and $i + 1$, and check that $|k + s_1 - s_2| = d_i$. If this is the case, make a transition to dp_{i+1,s_2-s_1}.
In the end we must have $dp_{n,d_n} = 1$ or $dp_{n,-d_n} = 1$.
Alternatively, one can show that the sequence D must satisfy the regexp $(0^*1[02]^*1)^*0^*$.

Complexity: $O(n)$.

D. Divide Into Thin Rectangles

Keywords: constructive, DFS and such.

For $k > 1$, any strip $1 \times k$ or $k \times 1$ is clearly tilable.
Consider a maximal (=non-extendable) strip s of length at least 2 on edge of the polyomino. Assume that the strip is horizontal, and only adjacent to cells below it.
Tile and remove s, and consider all maximal horizontal strips $s_1, \ldots, s_k$ it was adjacent to. Note that removing s splits the polyomino into k connected components, since it didn't contain any holes. Thus, we can consider the components of s_i independently.
If s_i has length 1, replace it with a maximal vertical strip down from s_i.
After this, s_i splits its component into two parts, both of which can be processed recursively.

To see why this procedure never fails, observe the following invariant: for each call with s and a respective component, all cells with a single neighbour must belong to s.
The invariant is evidently preserved after any call to s_i. The only possible issue is when s_i is vertical and has length 1. But then before the current call, it had a single neighbour, but wasn't in s, a contradiction.

Each cell will be considered $O(1)$ times across all subroutines (main recursion, finding maximal strips, etc), thus the **complexity** is $O(n^2)$.

E. Equilibrium of Colors

Keywords: dominator tree.

Add a source vertex s, and create edges from s to the first m vertices $w_1, \ldots, w_m$. Remove all vertices unreachable from s.
Construct the *dominator tree* of the resulting graph rooted at G. With induction on the dominator tree we can show that:

- If $u \neq s$ dominates v, then v must share the color with u.
- If only s dominates v, then v is either one of $w_1, \ldots, w_m$, or a mixer with a unique color.

Thus, the number of unique colors is the number of children of s in the dominator tree.

Complexity: $O((n + m + k) \log n)$ or $O((n + m + k)\alpha(n))$.

F. Fluctuation Values

Keywords: data structures.

Solution 1. Let $w_{l,r}$ be the sum in the subsegment $[l, r]$.
Process all queries and subsegments by increasing of weight, and maintain a two-dimensional data structure of $a_{l,r}$, where $a_{l,r} = w_{l,r}$ if $w_{l,r}$ below the current threshold, and $-\infty$ otherwise.
Each query (l, r) can answered with a 2D RMQ $\max_{i \geqslant l, j \leqslant r} w_{i,j}$. This has a sketchy **complexity** of $O((n^2+q) \log^2 n)$, but if 2D Fenwick tree is used, the constant factor should be fairly low.

Solution 2. Sort all subsegments $s_1, \ldots, s_{n(n+1)/2}$ by non-decreasing of weight. Maintain RMQ with values $l_1, \ldots, l_{n(n+1)/2}$, where l_i is equal to the left endpoint of s_i if s_i is "on", and $-\infty$ otherwise.
We now instead perform a sweep-line by increasing of the right endpoint of segments. For each right endpoint r:

- Turn "on" all subsegments with right endpoint r.
- For each query l, r, U locate the rightmost l_i such that $w(s_i) \leqslant U$ and $l_i \geqslant l$. The first condition turns it into a prefix query, which can be answered in $O(\log n)$ time with segment tree/treap/Fenwick tree descent.

This solution now has **complexity** $O((n^2 + q) \log n)$.

G. GPS for Robot

Keywords: simulation, planar graphs.

Subdivide all segments at pairwise intersections, creating a planar graph. We can now simulate the process directly. To avoid a possibly large number of loops, find the first time P we return to the starting point, and replace t with $t \bmod P$.

Complexity: $O(n^2 \log n)$ or $O(n^2)$.

H. Height Manipulations

Keywords: DP optimization.

For $j > i$, let $S(i,j) = (h_j + h_i) \cdot (j - i)$. One can see that this is twice the area of the trapezoid below the segment $(i, h_i) - (j, h_j)$.
If for $i < j$ we have $h_i \leqslant h_j$, then i is never an optimal right point. Similarly, if $h_i \geqslant h_j$, then j is never an optimal left point.
Removing unoptimal candidates, we have sets L and R of left and right candidates. L is an increasing chain, and R is a decreasing chain.
For any $i < j < k < l$ such that $i, j \in L$, $k, l \in R$ we can establish the *quadrilateral inequality*: $S(i,k) + S(j,l) > S(i,l) + S(j,k)$.
In particular, this implies $opt(r)$ — the optimal left endpoint position for the right endpoint r is non-decreasing.
Thus, we can apply D&C to solve the problem in **complexity** $O(n \log n)$. Apparently, $O(n)$ solutions also exist.

I. Industrial Robots

Keywords: math.

Suppose that robots are numbered $0, \ldots, n-1$ and are located at $x_0, \ldots, x_{n-1}$ respectively. The only suitable final arrangements are $X, X+d, \ldots, X+(n-1)d$ or $X, X-d, \ldots, X-(n-1)d$, for some choice of X.
In the first case, the maximum distance to arrive at the arrangement is $\max_{i=0}^{n-1} |x_i - (X + i \cdot d)| = \max_{i=0}^{n-1} |x'_i - X|$, where $x'_i = x_i - i \cdot d$.
By choosing X optimally at the midpoint between $\max x'_i$ and $\min x'_i$, the distance we get is $(\max x'_i - \min x'_i)/2$.
The second arrangement is treated similarly, and the best of two answers should be returned.

Complexity: $O(n)$. $O(n \log n)$ is fine too.

J. Jack the Black

Keywords: combinatorial probability.

Each of the 6×6 options has probability $1/36$, count them straightforwardly.

K. Kingdoms

Keywords: FFT/bitsets.

Given coordinates a_i, b_i, c_i on three layers, we need to count the number of triples (i, j, k) such that $b_j = (a_i + c_k)/2$. After a suitable shift, assume all coordinates are non-negative.
Consider the polynomial $A(x) = \sum x^{a_i}$, and $B(x)$ and $C(x)$ defined similarly. If we find $A(x) \cdot C(x) = D(X) = \sum_i d_i x^i$, then the answer is $\sum_j d_{2j} b_j$.
We can either use FFT to find $A(x) \cdot C(x)$, or use bitsets to find the answer as $\sum_k \text{popcount}((A << c_k) \& (2B))$.

Total complexity: $O(A \log A)$ or $O(nA/64)$, where A is the largest possible coordinate.

L. Lights On/Off

Keywords: Gaussian elimination.

Let $A_{ij} = 1$ if the switch j toggles the light i, and $V_{ij} = 1$ if we need to toggle the switch j if we only want the light i on.
We must have $AV^T = I$ modulo 2 (I is the identity matrix), thus $V = (A^T)^{-1}$.
The inverse matrix can be found with Gaussian elimination with bitset acceleration.

Complexity: $O(n^3/64)$. $O(n^3)$ is fine too.