A

A. Alice, Bob and Rectangles

You are given n rectangles $R_1, \ldots, R_n$ defined by their coordinates. Calculate the number of sequences $p_1, k_1, p_2, k_2, \ldots, p_n, k_n, p_{n+1}$, where k_i is an integer s.t. $1 \leqslant k_i \leqslant n$ and p_i is a point belonging to both $R_{k_{i-1}}$ and R_{k_i} (in case they exist).

A. Alice, Bob and Rectangles

Consider the following DP: $A[m][(x, y)]$ is the number of sequences ending in (x, y) if we consider first m rectangles of the original sequence. Of course we cannot store all DP values explicitly as there are too many of them, but we will try to investigate how these values behave.

A. Alice, Bob and Rectangles

What happens when we move from $m - 1$ to m?

First of all, when we introduce a new rectangle, all old sequences are valid, so old values of DP are kept as it is. All we need is to account sequences that end in $\ldots, p_{prev}, m, p_{cur}$, i.e. having m-th rectangle as a last rectangle of the sequence.

Note that constraints $p_{prev} \in R_m$ and $p_{cur} \in R_m$ are imposed.

So, for any $p_{cur} = (x, y) \in R_m$, $DP[m][(x, y)]$ should be increased by the same value of

$$add_m = \sum_{p_{prev}=(x',y') \in R_m} DP[m-1][(x', y')]$$

A. Alice, Bob and Rectangles

As far as presentation author knows/believes, there exist data structures that allow to perform online 2-d range sum and 2-d range increase/decrease queries in $O(poly(\log n))$ running time, but they are pretty hard and slow. So we will try to come up with a different approach of performing such operations.

A. Alice, Bob and Rectangles

Note that add_m consists of several summands introduced while processing rectangles from 1 to $m - 1$.

Key idea of our algorithm: we will try to apply those modifications in batch manner by grouping several modification rectangles and several recepient rectangles and running a sweep line algorithm to account effect of former ones onto the latter ones.

A. Alice, Bob and Rectangles

Namely, we will run a Divide-and-Conquer algorithm, which looks like a recursive function $process(l, r)$ with the following invariant: *when $process(l, r)$ is called, effect of all modifications from 1 to $l - 1$ onto rectangles from l to r is already accounted.*

In particular, by the moment of $process(l, r)$ the value of add_l is finalized.

If $l = r$, we do nothing.

If $r > l$, we consider $m = \left\lfloor \frac{l+r}{2} \right\rfloor$, then call $process(l, m)$, and then account effect of $l..m$ onto $(m + 1)..r$. Note that the order of these two operations matters!

A. Alice, Bob and Rectangles

In order to account effect of group of rectangles (let's call them blue) onto another group of rectangles (let's call them green), we should increase add_j for each green rectangle j by the sum of $add_i \cdot area(R_i \cap R_j)$ over all blue rectangles i.

This may be done by a line sweep keeping segment tree for range updates and range sum queries.

Such sweep works in $O(r \log r)$, where r is the total number of rectangles.

Total running time of such Divide-and-Conquer approach will be $O(n \log^2 n)$. The answer to the problem will be total equal to the total value of add_m across all rectangles.

B

B. Bugs

There are n points moving linearly in the plane. Calculate $\frac{1}{T} \int_0^T S(t)dt$, where $S(t)$ is an area of the convex hull.

B. Bugs

Consider a convex hull structure, which is a set of points that are vertices of convex hull.

Structure may change only in moments when some three points become collinear.

Three points $a(t)$, $b(t)$ and $c(t)$ change when oriented triangle area $area(a(t), b(t), c(t))$ turns to zero. Note that the last expression is a second-degree polynomial over t, so there are no more than 2 roots (if the expression is not a constant zero).

B. Bugs

Thus, there are $\Theta(n^3)$ interesting moments.

Calculate structure of convex hull on any moment-free segment by picking arbitrary point on it. Total running time of this part is $O(n^4 \log n)$.

On each moment-free segment, express hull area as a second-order polynomial of t and perform analytical integration of the desired expression.

C

C. Counting

There is a rectangular field $n \times m$ and k points inside it. Calculate number of rectangles inside field that contain at least one point.

C. Counting

We will count number of point-free rectangles instead.

Let's sweep a horizontal line from top to bottom. For each line location we will count the number of rectangles that are based on this line as a bottom side. Suppose line y-coordinate is y_0.

What happens when we move line from y_0 to $y_0 - 1$ without sweeping through any point?

C. Counting

Suppose that there were a rectangles based on y_0 and b of them were inextensible in upwards direction (i.e. they were right below some point or upper rectangle border).

For $y_0 - 1$ position of sweep line, all a previous rectangles may be shifted down by 1, and also there'll appear exactly b unextendable rectangles, so new values of a and b will be $a' = a + b$, $b' = b$.

So, between any two "interesting" locations of sweep line that pass through points, we have to sum values of some arithmetic progression, that is completely defined by a and b values of initial sweep line position.

C. Counting

How do we determine values of a and b for some sweep line position passing through the point?

Consider all kinds of inextensible rectangles. Each kind is defined by a point (or a rectangle border) that bounds it from above, which also defines its height. So, there are $O(k)$ inextensible kinds of rectangles; for kind consisting of x inextensible rectangles of height h we will add x to b and xh to a.

Finally, for each kind of inextensible rectangles their amount x may be found out by considering the sequence of heights based on current sweep line position, and using the "next smaller to the left" and "next smaller to the right" primitives utilizing stack.

Total running time will be $O(k^2)$.

D

D. Dracula's Garden

Given three points a, b and c on plane and n radii of circles, find out if it is possible to place circles of such radii so that three points become connected via circles.

D. Dracula's Garden

One can prove that if the answer is "possible", then there exists a point O such that it is possible to reach O from a, from b and from c walking along straightline segment. Moreover, all of the visited circles when starting from a, b and c are different except for the last circle containing O.

Let's iterate over all possible roles for all of the circles: there are $12 \cdot 3^{11} \approx 2 \cdot 10^6$ combinations.

For fixed role combination, O should belong to three circles of certain radii around a, b and c. Check if they intersect.

E

E. Editor

Implement a data structure maintaining sequence of numbers and a cursor, which allows to:

- insert a number after the cursor;
- delete a number before the cursor;
- move cursor by one position to the left or to the right;
- report maximum prefix sum to the left of the cursor.

E. Editor

Trying to implement some powerful generic data structure like balanced BST (treap, for example) or use offline nature of the problem and segment trees will, most probably, lead to TLE.

It is important to find out the restriction in the problem statement which allows us to implement a more efficient solution.

E. Editor

In this particular problem, the important restriction is that the cursor moves by one element to the left or to the right per operation.

Let's divide our problem into two parts:

- Structure A for maintaining some data structure for elements to the left from the cursor;
- Structure B for maintinaing a sequence of elelements themselves.

It is easy to see that structure B may be implemented using double-linked list with an iterator.

E. Editor

Structure A has a good property that underlying sequence of the numbers changes by no more than one element per operation when we add number, remove number of move the cursor by one position.

Implement it as a stack containing triples $(element, prefixSum, maximumPreviousPrefixSum)$.

F

F. Flow

Given a matrix of all pair maximum flows, determine if there exists a graph with such matrix.

F. Lamp

This problem requires knowledge of two deus ex machina-like theorems.

Theorem 1: for any graph G, there exists a tree T on the same set of vertices, which has the same all pair maximum flow matrix. Such relation between graphs is called *flow-equivalence*.

NB: T is not necessarily a Gomory-Hu tree; GH tree also requires that cutting any edge tree induces minimum cut between tree components in the original cut. For example, there are many flow-equivalent trees for $K_{1,3}$, but only one GH tree, which $K_{1,3}$ itself.

F. Lamp

Theorem 2: Maximum spanning tree in complete graph with $f(v, u)$ as cost function is a flow-equivalent tree for G.

These two theorems are closely related to theory of Gomory-Hu trees and their proof may be found in any related article.

So, the solution is: find maximum spanning tree of the complete graph with $f(u, v)$ as a cost function, and do not forget to check that this tree indeed satisfies the requirement, i.e. that $f(u, v)$ is equal to the minimum edge on the path between u and v. Overall complexity is $\Theta(n^2)$.

G

G. Group

Given a set of permutations $S \in S_n$, find the size of the group generated by S.

G. Group

This problem requires a direct application of Schreier-Sims algorithm. If we have enough time, we'll describe it on a blackboard. If not, just google it.

H

H. How Many Y-Triples

Calculate the number of tree vertex triples such that they cannot be covered by a tree path all together.

H. How Many Y-Triples

Each such triple is defined by a central vertex which is the intersection of pairwise paths between vertices. For given central vertex v, the number of Y-triples corresponding to it is the sum of all size products across all possible subtree triples of v, which can be calculated in linear by degree of v time, so the overall solution is linear.

I

I. Integer Sum

Given a positive integer $n \leqslant 10^{10^5}$, calculate the number of its expressions as an ordered sum of no more than n terms modulo $10^9 + 7$.

I. Integer Sum

It's easy to see that the number of expression as a sum is equal to 2^{n-1}.

2^{n-1} modulo $10^9 + 7$ is periodic with period of $10^9 + 6$ due to Fermat's little theorem.

So, we have to calculate $n - 1$ modulo $10^9 + 6$; this may be done while reading n digit by digit. Do so, then raise 2 to the power of the result to obtain the answer.

J

J. Japanese Olympic Center

You are given a map of connected area consisting of several connected units; cells with only one neighbour are called rooms.

You have K staffs that travel with certain speed and can check rooms in certain time. You have to check all rooms in all units by exactly one staff so that each unit is checked by exactly one staff in some order, and staffs move to new units only after checking all rooms in previous one.

Each staff starts in cell (s, t) and has to return there. What is the minimum possible maximum requried time across all staffs?

J. Japanese Olympic Center

This problem is pretty straightforward and requires a lot of careful implementation.

First of all, parse the input, run several DFSes and prepare list of units with their rooms. Assign each room a local index for its unit and a global index. Note that there are no more than $U = 12$ units with no more than $R = 12$ rooms in each of them.

J. Japanese Olympic Center

Define $UnitCheckTime[i][j][k]$ equal to the minimum check time of unit i if we start at room j and finish at room k. Here j and k are local indices of rooms j and k in unit i.

It can be calculated using intermediate DP: $UnitRoomsCheckTime[i][msk][j][k]$ equals to the minimum check time of rooms defined by mask msk in unit i such that we start at room j and finish at room k.

There are $O(U \cdot 2^R \cdot R^2)$ states for this DP, transition is done in $O(R)$, so the overall complexity is $O(U2^R R^3)$.

J. Japanese Olympic Center

Define $PathCheckTime[msk][i]$ equal to the minimum time to start at the initial cell O, then check mask msk of units and finish at the room i. Here i is the global index of the room.

It can be calculated via DP. For $msk = 2^k$ (k is a unit index):

$$PathCheckTime[msk][i] = \max_{j \in rooms[k]} \{dist(O, j) + UnitCheckTime[k][j][i]\}$$

For larger msk, iterate over previous unit and a room in it, and perform a transition.

There are $O(2^U \cdot RU)$ states, transition is done in $O(RU)$, so the overall complexity of this part is $O(2^U R^2 U^2)$.

J. Japanese Olympic Center

Define $TourTime[msk]$ to be the minimum time that is required to check all rooms in all units defined by mask msk. This value is easy to obtain by aggregating $PathCheckTime[msk][i]+dist(i, O)$.

Finally, calculate $CombinedTourTime[msk][k]$ which is the minimum possible maximum tour time while covering mask msk of units by k staffs. It can be done via DP which works in $O(3^U \cdot K)$.

Overall complexity of the solution is $O(U2^R R^3 + 2^U R^2 U^2 + 3^U \cdot K)$. Luckily, we do not have to recover the answer, so that's all, folks!