机房网络

考虑先把边反过来建，每个询问转化为从1号点走一条递增的路径到达x，且这条路径的第一条边的权值$x \geq l$，这条路径的最后一条边的权值$y \leq r$，我们记一条路径为(x,y)。

这时候考虑将边按照从小到大的顺序加入图中，记f_i表示当前状态下从1到点i的所有路径(x,y)中，x的最大值。那么当你加入一条边权为w的边$u \rightarrow v$时：如果u为1，你将会得到一条从1到v的路径(w,w)，然后将$f_v \leftarrow w$；否则如果f_u有值，你会得到一条从1到v的路径(f_u,w)，然后将$f_v \leftarrow f_u$。

因为一条边顶多贡献一条路径，所以你得到的路径总数是$\leq m$的。

之后对于每个询问(x,l,r)你只要查看是否有你处理出来的从1到路径x的路径(x,y)满足$l \leq x < y \leq r$，可以二维数点也可以拿$vector$边处理路径边处理答案解决。

注意边权可能相同，边权相同的边转移的时候要一起处理。

复杂度$O(n \log n)$，瓶颈在排序。

数中分

$sub0$

$Sol1$ 查看 `jntm3.in` 后发现输入是哥哥帅气的照片，易证他的生日是 1998年8月2日，故输出19980802。

$Sol2$ 查看 `jntm3.out` 后发现答案是19980802，故输出19980802。

$O(1)$

$sub1$

枚举其中一条背带，再用前缀和检查中分的范围内是否全是1。$O(n^4)$

$sub2$

先 $O(n^2)$ 枚举两条背带的交点，然后用单调栈维护对于这个交点:

1. 以它作为背带的右下角，背带的宽为 x，背带的最大高度，记为 $f(x)$。
2. 以它作为背带的左上角，背带的高为 x，背带的最大宽度，记为 $g(x)$。

则这个交点的贡献为 $\sum_x \min(f(x),g(x))$。

写法上可以注意本质不同的中分只有两种，所以可以先跑一遍，把原矩阵镜像之后再跑一遍。

$O(n^3)$

$sub3$

考虑用数据结构维护上述操作，但难点是处理完 (x,y) 的信息后，这些信息对于 $(x,y+1)$ 和 $(x+1,y)$

会产生错位。

容易想到按斜线枚举交点，即处理完 (x,y) 后，下一个处理 $(x+1,y+1)$ 或 $(x-1,y-1)$。

以下一个处理 $(x-1,y-1)$ 为例：考虑从 (x,y) 转移到 $(x-1,y-1)$ 的过程中，发生了什么变化。

其中 f 数组经历以下三个变化：

1. 由于向左移了一位，故删除数组最右边的一项 x 。若此时 右边第二项 大于 x ，则它在经历了暗无天日的磨难后终于迎来了出头之时。需要检查后面的项是否能造成更多贡献，并更新贡献，即区间赋值。
2. 由于向下移了一位，故 f 的高度整体加一，即全局加。
3. 向下平移后拓展到的这一行可能并不全都是1，其中最靠近右侧的0会将左边全部截断，即区间赋值。

变化一的复杂度可能有疑问，但事实上遍历整个矩阵后，变化一的更新次数恰好等于单调栈中弹出的总项数，故不超过 n^2 次。

实现上可以先预处理出单调栈，单调栈的前后两项连边，实际上变化一需要更新的项就在 右边第二项 到 右边第一项 的路径上。

问题转化为一棵维护两个单调不降数组的线段树，支持区间赋值、全局加一、以及求 $\sum_{i=1}^{n}\min(f_i,g_i)$ 。

记录下每个节点管辖范围的 $f_{min},g_{min},f_{max},g_{max},f_{sum},g_{sum},ans$ ，区间赋值时当节点的 $[f_{min},f_{max})$ 和 $[g_{min},g_{max})$ 无交时即可更新答案。

由于 f,g 均为单调不降函数，所以单次区间赋值是 $O(\log n)$ 的。

总复杂度 $O(n^2\log n)$ ，常数由于操作过于繁琐相对较大。

出题人的代码常数有亿点大，且已经尽可能地卡掉 $O(n^3)$ 做法了，如果还有卡常暴力过了这题，谢罪。

摆阵

令 $V=\max a_i$。

根据矩阵树定理，我们要求 $\det(D-G)=\det(D-AA^T)$，其中 $A_{i,j}=a_i[j|a_i]$。

由 Matrix determinant lemma，有 $\det(D-AA^T)=\det(D)\det(I-A^TD^{-1}A)$。

前面一项是好求的，后面的我们考虑先把矩阵求出来。

令 $B=A^TD^{-1}A$，则有 $B_{i,j}=\sum\limits_{k=1}^{n}\dfrac{[i|a_k][j|a_k]a_k^2}{deg_k}$。这个式子只与 $\operatorname{lcm}(i,j)$ 有关，可以预处理后计算。

容易发现当 $\operatorname{lcm}(i,j)>V$ 的时候，$B_{i,j}=0$，且 i 越大该行不为 0 的项越少。因此对这个矩阵按行的编号由大到小进行高斯消元即可，实际运行效率极快。若按行的编号由小到大进行高斯消元只能过 $V\le 500$。