牛客竞赛
AC.NOWCODER.COM

2024 牛客
暑期多校训练营
第二场

qcjj 粉丝团

HR.NOWCODER.COM

A - Floor Tiles

题目大意

- 构造一个 $N \times M$ 的平面，包含两种类型的地板
- 地板边缘中心点相连的曲线合并成一条曲线
- 问能否构造出仅包含恰好 K条曲线的地板
- $1 \leq N, M \leq 800, 1 \leq K \leq 2 \cdot N \cdot M$

A - Floor Tiles

题解

- 注意到无论 N, M 是多少，平面的结构如何，从整个二维平面边缘进入的曲线最终一定会从整个二维平面边缘出去，不可能进入其内部的任何一个环
- 边缘的“线头”一共有 $2(N+M)$ 个，所以这样的曲线数目为 $\frac{2(N+M)}{2} = N+M$ 个
- 所以对于一个大小为 $N \times M$ 的平面，其曲线总数为 $N+M+$ 平面结构内部环的数目

A - Floor Tiles

题解

- 接下来只需要考虑如何造环，首先可以没有环，可以使用全 A 或者全 B 的方式很容易构造
- 那怎么造出特定有 K 个环的结构呢，先考虑如何造最多的环
- 显然，构造单位圆 $\begin{matrix} A & B \\ B & A \end{matrix}$ 这种方式是最优的
- 在最优情况下，如果左上角为 A 类型，则最多存在 $N+M+\left\lfloor\frac{(N-1)(M-1)+1}{2}\right\rfloor$ 个曲线；如果左上角为 B，则最多 $N+M+\left\lfloor\frac{(N-1)(M-1)}{2}\right\rfloor$ 个曲线；无论如何，最少都为 $N+M$ 个曲线

A - Floor Tiles

题解

- 构造方法也很简单，首先根据固定的一块，按照相邻相反贪心得到左上角地板的类型，以及可行解的答案区间
- 如果有解，根据左上角类型按照 AB 交替的方式填，构造最多的单位圆
- 然后从上到下，从左到右开始全填 A 或者 B，覆盖掉单位圆，直到满足条件

B - MST

题目大意

- 给定一个 n 个点的简单带权无向图 G
- 每次询问一个点集 S，求 S 关于 G 导出子图的最小生成树
- 没有输出 -1
- $2 \le n \le 10^5, 1 \le m, q \le 10^5, 1 \le w_i \le 10^9, \sum k_i \le 10^5$

B - MST

解法 1

- 因为没有办法使用邻接矩阵，所有首先用 map，将整个图存下来
- 考虑两种可能的暴力
- 第一种：对于给定点集 S，双重循环枚举 S 中的每个点，把其中有效的边均取出来，并排序
- 使用 kruskal 算法，可以求出最小生成树，整体时间复杂度为 $O(n^2\log(n))$

B - MST

解法 1

- 第二种：对于给定点集 S，循环枚举 S 中的每个点，在循环枚举该点的所有边，将合法的边取出来，并排序
- 使用 kruskal 算法，可以求出最小生成树，最坏情况下整体时间复杂度为 $O(m\log(n))$
- 可以发现对于 $base = \sqrt{n}$ 的时候，如果使用第一种暴力，那么单次时间复杂度为 $O(|S|^2\log(n))$

B - MST

解法 1

- 对于 $|S| > base$ 时，使用第二个算法，那么最坏时间复杂度为 $O(m\log(n))$ 的
- 由于第一个算法的 $|S| \le base$，所以所有第一个暴力的时间之和的复杂度为 $O(n \cdot base \cdot \log(n))$
- 由于第二个算法的 $|S| > base$，所以所有第二个暴力的时间之和的复杂度为 $O(base \cdot m\log(n))$
- 根号分类即可

B - MST

解法 2

- 考虑如何快速筛选出需要的边，并且不多枚举
- 由于存在类菊花图的可能，所以考虑如何降低，边数多的点所存的边
- 考虑三元环计数的方式，将边存成单向边，那么对于每一个点的边数，都会被降低到 $\sqrt{m}$
- 枚举 $|S|$ 中的每一个点，枚举他的所有边，此时时间复杂度为 $O(|S|\sqrt{m}\log(|S|\log(n))$

B - MST

解法 3

- 由于度数大于 $\sqrt{m}$ 的最多 $\sqrt{m}$ 个点，所以首先对点做分类
- 将点分为两类，1: 度数大于 $\sqrt{m}$ 的点，2: 度数小于的点
- 考虑对于 1 类点，边数会比较多，所以考虑 1 类点通过邻接矩阵来存
- 对于 2 类点，边数会比较少，考虑 2 类点通过邻接表来存
- 那么整体的空间复杂度为 $O(n\sqrt{m})$

B - MST

解法 3

- 考虑如果查询 $|S| \le \sqrt{10^5}$
- 由于朴素的 Prim 算法，不使用堆优化的情况下，单次复杂度为 $O(|S|^2)$
- 对于 1 类点，每次更新边时可以直接枚举 S 中的每个点，由于是邻接矩阵，所以整体时间复杂度为 $O(|S|^2)$
- 对于 2 类点，每次枚举整个数组，将符合集合 S 的边提取出来，由于 2 类点度数小于 $\sqrt{m}$，所以整体时间复杂度为 $O(|S|\sqrt{m})$
- 整体时间复杂度为 $O(|S| \times (|S| + \sqrt{m}))$，由于 $|S| \le \sqrt{m}$，所以整体时间复杂度为 $O(|S|\sqrt{m})$

B - MST

解法 3

- 如果查询 $|S| > \sqrt{10^5}$，考虑对于生成树会包含什么样的边，一定是 1 类点连接 2 类点，1 类点连接 1 类点，2 类点连接 2 类点
- 对于 1 类点连接 1 类点，直接双重循环枚举即可，复杂度最坏为 $\sqrt{10^5}^2$
- 对于 1 类点连接 2 类点，考虑先枚举 2 类点，枚举 2 类点所连接的边是否连接 1 类点，由于 2 类点的度数，整体复杂度为 $O(m)$
- 对于 2 类点连接 2 类点，同上
- 将所有可能的边丢入对应的邻接表中，整体复杂度为 $O(m)$

B - MST

解法 3

- 考虑同样使用 Prim 算法，总共会更新点权最多 m 次，找到最小值 $|S|$ 次
- 如果使用不优化，复杂度为 $O(m+|S|^2)$，如果使用堆优化，复杂度为 $O((m+|S|)\log(|S|))$
- 均会使时间复杂度大于等于 $O(m^{\frac{3}{2}}\log\sqrt(m))$
- 考虑更换数据结构维护最小值，使用分块，每次维护一个块的最小值

B - MST

解法 3

- 当块中出现一个新的值时，整个块的最小值会被直接更新，单次复杂度为 $O(1)$
- 找到最小值，由于最多存在 $\sqrt{n}$ 个块，所以找到最小值的单次复杂度为 $O(\sqrt{n})$
- 找到最小值之后，该块无法知晓最小值是谁，可以暴力重构该块，复杂度为 $O(\sqrt{n})$
- 整体复杂度为 $O(m+|S|\sqrt{n})$
- 由于 $|S|>\sqrt{m}$ 的次数一定是小于 $\frac{10^5}{\sqrt{10^5}}$ 的，全局复杂度在最大情况下为 $o(n\sqrt{n})$

B - MST

解法 4

- 考虑如果查询 $|S| \le \sqrt{10^5}$
- 直接 Prim
- 如果查询 $|S| > \sqrt{10^5}$ 那么边数可能会大于 m，提前对所有边排序
- 每一次查询直接遍历所有边，如果两个端点都属于，那么为合法点，使用 Kruskal 算法，将并查集使用上路径压缩 + 按秩合并
- 整体时间复杂度为 $O(n\sqrt{n})$

C - Red Walking on Grid

题目大意

- 给定一个 2 行 n 列的地图，有一些格子为障碍
- 任选起点，每个格子最多只能走一次（不能走到障碍），求最长路径
- $1 \leq n \leq 10^6$

C - Red Walking on Grid

题解

- 贪心
- 在第一行时，优先向下去第二行，其次向右。在第二行时，优先向上去第一行，其次向右
- 如果某次向下后右侧为障碍，则此次向下取消，直接向右；向上同理
- 考虑到奇偶的特殊性，因此当同一列有两个障碍时，起点从该障碍右边第一行还是第二行开始都需要考虑，因此需要枚举两次
- 复杂度为 $O(n)$

D - Taking Candies

题目大意

- A, B 两个人抢 n 盒糖果，第 i 盒糖果有 a_i $(1 \le i \le n)$ 颗糖
- n 次竞价，每次竞价的过程是：
  - 假设一开始 A, B 分别有 p, q 个硬币
  - A 先手出价 $0 \le v_0 \le p$
  - 然后 B 出价 $v_0 < v_1 \le q$ 或弃权
  - 然后 A 出价 $v_1 < v_2 \le p$ 或弃权
  - ...
  - 直到某个人弃权，假设最后出价是 v_m，则出价的人将 v_m 个硬币给弃权的人，并且出价的人可以随意拿走一盒糖果

D - Taking Candies

题目大意

- 一开始 A, B 分别有 x, y 个硬币，问最优策略下 A 最多拿多少糖果
- $1 \leq n \leq 10^5, 0 \leq x, y \leq 100$

D - Taking Candies

题解

- 观察 1：拿糖果的顺序是固定的，一定是按 a_i 降序的顺序拿
- 观察 2：一定存在一个最优策略，A,B 出价的次数不超过 1 次
- 观察 3：硬币总数是 $s = x + y$ 在整个过程中是固定的

D - Taking Candies

题解

- 设 $g(i, t)$ 是倒数第 i 轮 A 取得 t 个糖果所需的最少硬币
- 假设 A 有 x 个硬币，B 有 $s-x$ 个硬币，这一轮的糖果数是 w
- 如果 A 想得到这轮的 w 个糖果，他最多可以出的价是 $u = x - g(i-1, t-w)$
- B 可以出价 $u+1$，虽然 A 得不到糖果，但只要 $x+u+1 \geq g(i-1, t)$，A 之后还是可以得到 t 个糖果
- 得到

$$2x + 1 - g(i-1, t-w) \geq g(i-1, t)$$

D - Taking Candies

题解

- 于是

$$g(i,t)=\lfloor\frac{g(i-1,t-w)+g(i-1,t)}{2}\rfloor$$

- 虽然 t 很大，但 $g(i,t)$ 本身只能有 $s+2$ 个取值（$0..s+1$，其中 $s+1$ 表示不可能取到），并且 $g(i,t)$ 关于 t 是单调的，维护每个取值对应 t 的范围即可
- 复杂度：$O(n(x+y))$

E - GCD VS XOR

题目大意

- 给定 x，构造 $y < x$ 使得 $gcd(x, y) = x \oplus y$
- $1 \leq x \leq 10^{18}$

E - GCD VS XOR

题解

- 取 $x - \mathrm{lowbit}(x)$ 即可，如果 x 是 2 的整数次幂则无解。
- 单组 $O(1)$

F - Mixed Juice

题目大意

- 动态维护若干个二维限制，每次查询是否存在从每个限制的范围内取一点，横纵坐标分别求和恰好等于某个二维点的横纵坐标。
- $1 \le N, Q \le 2 \times 10^5$
- $1 \le x \le N, 1 \le l_i \le r_i \le 10^9, 1 \le L_i \le R_i \le 10^9$
- $1 \le s_i \le t_i \le N, 1 \le W_i, V_i \le 10^{18}$

F - Mixed Juice

题解

- 首先假设只使用一种果汁制作混合饮料，假设 $k_i = 1$，那么此时能够制作出的产物 $P(W, V)$ 在二维平面上的可行范围，应该是一个 $W \in [l_i, r_i], V \in [L_i, R_i]$ 的矩形
- 更进一步的，当 $k_i = 0.5$ 时，发现产物 $P(W, V)$ 在二维平面上的可行范围仍然是一个矩形范围，只不过矩形的位置和大小不同

F - Mixed Juice

题解

- 想象枚举 $k=[0,1]$ 的所有实数，这些矩形区域在二维平面上的可行范围全部叠加到一起，得到了一个纺锤四边形
- 即单点查询实际上就是问点 $P(W,V)$ 是否在四边形内
- 更进一步，考虑查询一个范围的区间，纺锤四边形是一个凸包结构
- 题目要求分别求和，相当于求凸包的闵可夫斯基和，查询问题转化成问点 $P(W,V)$ 是否位于 $[s,t]$ 构成的闵可夫斯基和凸包内

F - Mixed Juice

题解

- 然后因为题目要求带修改，直接树套树维护凸包实际上是不好搞的
- 这里处理的不好可能写成 $n\log^3(n)$ 的算法，实际上存在一种很优雅的转化
- 考虑将凸包按照上下两个半平面切分，分成上下两个半凸壳
- 如果查询点位于下凸壳的上方，并且位于上凸壳的下方，满足这两个条件，则点在整个凸包内部

F - Mixed Juice

题解

- 为了判断点是否位于半凸壳的上方/下方，需要找到查询点正上方/下方是哪一条边，它的坐标和斜率是多少。
- 在本题中，由于 w, v 两种属性都是正整数，所以前缀和 $\sum w, \sum v$ 单调递增
- 而前缀和 $\sum w, \sum v$，恰好就是凸壳上每一条边的起点坐标
- 对于一个半凸壳，其斜率具有单调性，所以可以构造一个起点坐标关于斜率的单调函数

F - Mixed Juice

题解

- 接下来就可以二分一个“第 k 大”的斜率，看小于/大于这个斜率的边求前缀和后，接下来的一条边是否是“恰好”落在查询点正上/下方的边，如果是就计算答案，否则继续二分
- 整个过程就是“带修改的第 k 大查询”，转化到了这一步，外面直接套一个整体二分的板子，用树状数组求前缀和 $\sum u, \sum v$，这个题就做完了，并且由于整体二分天然支持修改，比树套树好写了一万倍
- 时间：$O((n+q)\log^2(n+q))$，空间：$O(n+q)$

G - The Set of Squares

题目大意

- 定义一个多重数集合是好的，但且仅当集合中元素乘积是一个正整数的平方，集合的权值为这个正整数的值。
- 求大小为 N的多重数集的所有子集中所有好集的权值和
- $1 \le N \le 1000, 1 \le S_i \le 1000$

G - The Set of Squares

题解

- 注意到 $\sqrt{1000}$ 以内的质数只有 $\{2,3,5,7,11,13,17,19,23,27,31\}$ 一共 11 个
- 注意到对于 1000 以内的数，质因子中不可能出现两个大于 32 的质数
- 按照大质数分组，小质数状态压缩当成背包的体积，做分组背包
- 注意：数字 1 根据写法不同，可能要特殊处理，大质数配对、小质数配对都需要计算贡献，分类讨论不要遗漏情况
- $O(2^{11} \times N)$

H - Instructions Substring

题目大意

- 平面直角坐标系中，小红初始站在原点
- 给定一个包含“上、下、左、右”的、长度为 n的指令序列，以及一个特殊点 (x, y)
- 求选定一个子串，小红根据该子串序列移动，可以经过特殊点的方案数
- $1 \leq n \leq 2 \times 10^5, -10^5 \leq x, y \leq 10^5$

H - Instructions Substring

题解

- 前缀和 / map
- 枚举子串的左端点，对于“第一次经过特殊点 (x, y) ”的右端点，该右端点右边均为合法的右端点
- 倒着用 (unordered) map 存前缀和数组的值和下标，找差分的下标即可
- $O(n)$ 或 $O(n\log(n))$

I - Red Playing Cards

题目大意

- 给定一个长度为 $2 \times n$ 的数组，1 到 n 每个元素恰好出现两次
- 每次操作可以删除一个长度不小于 2 的连续子数组，需要满足该子数组首尾相同，获得该连续子数组“首尾元素值”乘以“元素数量”的分数
- 问最终可以得到的最大分数
- $1 \le a_i \le n \le 3000$

I - Red Playing Cards

题解

- 设 $1, 2, \ldots, n$ 的位置分别是 $(l_1, r_1), (l_2, r_2), \ldots, (l_n, r_n)$
- 设 $f(i)$ 是区间 $[l_i..r_i]$ 的答案
- 可以先假设 $[l_i..r_i]$ 的每个数的贡献都是 i
- 如果碰到另一个区间 $[l_j..r_j]$ 满足 $l_i < l_j < r_j < r_i$，那么再用 $f(j)$ 来替代 $[l_j..r_j]$ 这个区间的贡献

I - Red Playing Cards

题解

- 也就是令 $g(k) := [l_i..k]$ 当前的最大贡献
- $$g(k)=\begin{cases}\max\left(g(k-1)+i,\, g(l_j-1)+f(j)\right), \text{if } k=r_j \text{ and } l_j>l_i\\ g(k-1)+i,\ \text{otherwise}\end{cases}$$
- 根据定义，$f(i)=g(r_i)$
- 按长度 r_i-l_i+1 从小到大来计算 $f(i)$ 即可
- 一个 trick 是在 a 两端补 0，答案就是 $f(0)$
- 复杂度为 $O(n^2)$

J - Involutions

题目大意

- 集合 A 上的映射 $f\colon A \to A$ 为 involution，当且仅当任意 $x \in A$，$f(f(x)) = x$
- f 的不动点数为 $FP(f) = |\{x \in A | f(x) = x\}|$
- f 的权值为 $w(f) = a^{FP(f)}$，其中 a 是给定常数
- 求 $\{1, 2, ..., n\}$ 上的所有 involution 的权值和，答案对 998244353 取模
- $1 \le n \le 5 \times 10^9$

AC.

牛客竞赛 AC.NOWCODER.COM

J - Involutions

题解

- 省流版： $g(n) = ag(n-1) + (n-1)g(n-2),\ g(0) = g(1) = 1$
- 套一个整式递推模版即可

AC.

J - Involutions

题解

- 详细版：
- 设 $g(n) := \{1, 2, ..., n\}$ 上的所有 involution 的权值和
- 考虑一个 involution f 的 $f(n)$ 是 n 还是 $1..n-1$
- $g(n) = ag(n-1) + (n-1)g(n-2),\ \ g(0) = g(1) = 1$
- 令 $u_i = [g(i), g(i+1)],\ M_i = \begin{pmatrix} 0 & i+1 \\ 1 & a \end{pmatrix}$
- 则 $u_{i+1} = u_i M_i$

J - Involutions

题解

- $u_{n-1} = u_0 M_0 M_1 M_2 \ldots M_{n-2}$
- 设 $f_d(x) = \prod_{i=0}^{d-1} M_{x+i}$
- M_i 中各项关于 i 的次数不超过 1，$f_d(x)$ 中各项关于 x 的次数不超过 d
- 分块，设块长为 s，$t = \lfloor \frac{m}{s} \rfloor$，则

$$f_m(0) = f_s(0) f_s(s) \ldots f_s((t-1)s) f_{m-ts}(ts)$$

J - Involutions

题解

- 倍增的求 $f_s(0), f_s(s), ..., f_s((t-1)s)$
- 对于 $f_d(x)$，需要维护 $d+1$ 个点值 $f_d(ks)$ $(0 \le k \le d)$
- $f_{2d}(ks) = f_d(ks)f_d(ks+d)$，$f_{d+1}(ks) = f_d(ks)M_{ks}$
- 用点值平移（Langrange 插值）就可以 $O(d\log(d))$ 的从 f_d 得到 f_{2d} 的点值
- 暴力可以 $O(d)$ 的从 f_d 得到 f_{d+1} 的点值
- 于是可以 $O(s\log(s))$ 的得到 $f_s(0), f_s(s), ..., f_s((t-1)s)$

J - Involutions

题解

- 最后需要把矩阵乘起来，复杂度是 $O(n/s)$
- 令 $s=\sqrt{n}$
- 最终复杂度是 $O(\sqrt{n}\log(n))$

THANKS!

AC.NOWCODER.COM