团队

2020.08.03 2020牛客暑期多校训练营（第八场） `pro: 5/7/11` `rk: 10/685`

2020.08.01 2020牛客暑期多校训练营（第七场） `pro: 8/8/10` `rk: 5/1090`

个人

zzh

专题

比赛

题目

pmxm

专题

比赛

题目

jsh

比赛

- 2020/7/24 SRM 788 `problems: 1/2/3` `rank: 112/190`
- 2020/7/24 Codeforces Round #659 (Div. 1) `problems: 2/2/6` `rank: 220/1169`
- 2020/7/25 ABC: M-SOLUTIONS Programming Contest 2020 `problems: 5/6/6` `rank: 281/6527`
- 2020/7/29 Educational Codeforces Round 92 (Rated for Div. 2) `problems: 6/6/7` `rank: 55/13826`
- 2020/7/30 Codeforces Round #660 (Div. 2) `problems: 5/5/5` `rank: 24/13083`

专题

无

题目

无

本周推荐

zzh

XVI Open Cup named after E.V. Pankratiev. GP of Ukraine J. Joining Powers

Tagsnumber theory, binary search

题意：见链接

题解：见链接

Comment：有点意思的数论，如果你看着指数最多 60，想着去暴搜，你就输了。

pmxm

Atcoder M-SOLUTIONS Programming Contest 2020 problem E

https://img.atcoder.jp/m-solutions2020/editorial.pdf

Tagssearch

题意：平面上若干个点 已在x轴，y轴建设道路。定义点权重为点上的人数和到任意一条道路的最小距离之积。问假设可以再增加k条平行于x或y轴的道路。求最小点权和。

题解： 一个简单的想法是暴力枚举在哪些点上建设道路 然后check这样的复杂度是2^n * n^3的。其实预存一张表来记录每个点建设道路后的对其他点的更新，这样check的复杂度是n^2的。 实际上可以3^n去枚举所有可能的情况(无道路，平行x轴，平行y轴。这样是3^n * n的。

Comment：考虑清楚3^n,2^n * n^2 等复杂度和如何优化查询

jsh

Codeforces Round #660 (Div. 2) - E. Uncle Bogdan and Projections

题目链接

Tags：几何Convex Hull Trick

题意：有 $1 \le n \le 2\,000$ 条在 x 轴上方、平行于 x 轴的线段。现在你可以让这些线段以相同的一个方向平移到 x 轴上，就像一个平行光源打到了这些线段上，然后在 x 轴上投影。**但是要求投影之间不能相交**。记投影的范围为最大坐标和最小坐标的差，求投影之间不相交情况下，最小的投影范围是

多大。

题解

首先，如果线段都在同一个水平线上，那投影的范围是不会变的。所以以下我们考虑存在有至少一对线段，所在高度是不同的。

记投影方向与 y 的负半轴的角度为 $\theta \in (-\pi, \pi)$容易得出点 (x, y) 的投影的横坐标为 $f_{x, y}(\theta) = x + y \tan{\theta}$

让 $u = \tan{\theta} \in \mathbb{R}$改写一下函数 $g_{x, y}(u) = x + y u$题目需要的就是在投影之间不相交情况下，最小化所有线段上的点的函数值的最大值和最小值的差，相当于线段端点的函数值的最大最小差。

一个显然的想法是取线段投影没有相交、但是存在几个投影恰好相切时候的角度来对答案进行更新。

因为在投影之间完全不相交也不相切的情况下，角度逐渐偏左或偏右会让投影之间变化到恰好相切，根据 $g_{x, y}(u)$ 容易知道在取恰好相切的时候是能够取到答案的局部极值点的（极大或极小都有可能，但总有一个极值点比所有没有相切的情况要优）。

好，接下来我们需要能够知道，在什么情况下，才不会有投影相交。

容易发现一对高度不同的线段，会在某个特定的角度区间出现投影相交的情况，而求出来区间的左右端点刚好是这两个线段投影相切时候的角度。那我们 $\mathcal{O}(n^2)$ 处理出来所有的区间（可以用分数类），让端点为关键点，排个序 $\mathcal{O}(n^2 \log n)$然后在扫描线过程中不计算存在有投影相交情况下的关键点即可。

你会想说单关键点计算时间复杂度 $\mathcal{O}(n)$ 不就爆了吗。

会发现 $g_{x, y}(u)$ 都是些直线。应用一下 Convex Hull Trick，我们就能在 $\mathcal{O}(\log n)$ 的时间内计算 $\mathcal{O}(n)$ 个线性函数在某个自变量下的最大值啦。开两个表，一个 g一个 $-g$在关键点各自取最大值求和即可。

时间复杂度 $\mathcal{O}(n^2 \log n)$

Comment：为数不多能现场 AK 的比赛，写一个 E 题的题解纪念一下。