

JZPFAR解题报告

南京师范大学附属中学 顾昱洲

Contents

1	题目大意	2
2	50%的算法	2
3	100%的算法	3

1 题目大意

平面上有 n 个点。有 m 次询问，每次给定一个点 (px, py) 和一个整数 k ，询问 n 个点中距离该点第 k 大的点的标号。若有多个点距离该点相同，那么认为标号小的点距离较大。

50%的数据， n 个点的坐标在某范围内随机分布。

100%的数据， $n \leq 10^5$ ， $m \leq 10^4$ ， $1 \leq k \leq 20$ ，所有点的坐标满足绝对值 $\leq 10^9$ ， n 个点中任意两点坐标不同， m 个询问的点的坐标在某范围内随机分布。

2 50%的算法

首先，对于 $k = 1$ 的情况，我们可以得到一个定理：

定理1. 求出给定的 n 个点的凸包 H ，离某个点 P 的最远点必然是 H 的某个顶点。

证明 若不然，我们设最远点为 S 。由于 S 在 H 内，因此连接并延长 PS ，延长线必然与凸包的外轮廓相交。

若交点是凸包的一个顶点，那么这个顶点到 P 的距离比 S 大，矛盾；

否则，设交在某条边上，这条边的两个端点分别是 A 和 B ，交点为 O 。不妨假设 $\angle POA \geq 90^\circ$ ，那么根据正弦定理，必然有 $|PA| > |PS|$ ，矛盾。

证毕。

类似的结论可以扩展到 k 远点。为了方便表述，我们定义 p 层凸包：

定义1 (p 层凸包的递归定义). 1层凸包为所有点的凸包。

p 层凸包为删去前 $p - 1$ 层凸包的所有顶点后，剩下的点的凸包。

那么，有定理：

定理2. 求出给定的 n 个点的前 k 层凸包 H_1, \dots, H_k ，离某个点 P 的 k 远点必然是 H_1, \dots, H_k 中的某个凸包的顶点。

证明和定理1差不多，这里就不赘述了。

那么，我们暴力求出前 k 层的凸包，然后对于每个询问，对于前 k 层凸包上的每个点求出距离，然后暴力求第 k 大即可。

但是这里还存在一个问题。即前 k 层凸包上的点的总数。

注意到这里的关键条件：给定的 n 个点的坐标在某范围内随机分布。

对于1层凸包，我们有一个定理 [1](事实上，这篇论文证明的结论比这个定理要强得多)：

定理3. 在 \mathbb{R}^2 上随机 n 个点，那么这 n 个点的凸包上的点的个数的期望是 $O(\sqrt{\ln n})$ 的。

对于前 p 层凸包，我猜测可能会有类似的结论，不过可能会相当复杂。不过实际情况是， 10^5 个随机点的前20层凸包上总共有二千多个点。这是一个相当小的数。

因此我们暴力即可。

每次暴力求凸包时间复杂度 $O(k_m n \log n + Lm)$ ，其中 k_m 为 k 可能达到的最大值，本题中为20， L 为前 k_m 层凸包上点的个数。事实上，经过适当优化，复杂度可优化为 $O(n \log n + k_m n + Lm)$ 。不过求前 k 层凸包在这里不是占用时间的主要矛盾。

这里还有两点需要注意：

1. 由于随机的性质，求前 k_m 远点，几乎所有情况下均只需求前10层凸包。这个性质可以大大减少时间常数。

2. 一个数列求 k 大值，C++选手可以使用STL中的`nth_element`，时间复杂度是线性的，可以在一定程度上减小编程复杂度。

3 100%的算法

考虑和求 k 远点相近的一个问题：求 k 近点。再将之降低难度，变为求最近点。这个问题该怎么做？

一个熟知的方法是求出给定的 n 个点的Voronoi图 [2]，然后进行点区域查询。时间复杂度是 $O(n \log n + m \log n)$ 。这个方法是可以扩展到这里要解决的问题的，时间复杂度为 $O(k_m n \log n + m \log n)$ ，不过这个方法对于OI来说编程复杂度过大，且常数不一定能够承受，因此这里不作介绍。

这里介绍一个更加简单的方法：使用二维k-d tree来解决这个问题。集训队员应当已经知道这些内容。不知道其为何物，或如何构造，或如何使用它来解决最近点问题，请参考 [3]。

使用二维k-d tree解决 k 远点，是求最近点的方法的自然拓展。我们只需要维护一个长度为 k 的列表，存储当前的 $1 \dots k$ 远点。并且在访问一个结点的两个孩子时，优先访问可能使得结果变得更优的孩子。

一次询问的时间复杂度最坏显然是 $O(kn)$ 的(考虑 n 个点都在一个圆上，而询问的点在圆心)，期望可能是 $O(k\sqrt{n})$ 。实际测试的结果大致符合这个复杂度。但是我不会证明这个结论。注意这里使用了询问的点在某范围内随机这个关键条件。

由于不是占用时间的主要矛盾，我们可以暴力建二维k-d tree。

总的时间复杂度可能是 $O(n \log^2 n + mk_m \sqrt{n})$ 。

参考文献

- [1] Hueter, I., Limit Theorems for the Convex Hull of Random Points in Higher Dimensions, *Trans. Amer. Math. Soc.*, **351**, 4337-4363, 1999
- [2] Voronoi, G., Nouvelles applications des paramètres continus á la théorie de formes quadratiques, *Journal für die reine und angewandte Mathematik*, **134**: 198 - 287, 1908
- [3] http://en.wikipedia.org/wiki/K-d_tree