Catch The Penguins

杭州学军中学 张闻涛

1 问题简述

给出四维空间上N个企鹅坐标。

有Q个询问，每次给出一个点，问有多少企鹅每一维坐标都小于等于这个点。

样例输入:

1
0 0 0 0
2
1 1 1 1.0
1 1 1 -1

样例输出:

1
0

数据范围

共20个数据
数据1~2 N,Q≤5000
数据3~6 N,Q≤10000
数据7~14 N≤30000,Q≤10000
数据15~18 N≤10000,Q≤30000
数据19~20 N,Q≤30000

1.1 算法0

直接暴力枚举，对于每一个询问，一个for循环枚举每只企鹅，判断四维坐标是否都小于等于询问的坐标。

时间复杂度O(n^2)

期望得分10分。

1.2 算法0优化

把企鹅分别按四维坐标排序

对于每个询问，每一维在排好序的这四个数组里二分得出这一维小于询问坐标的企鹅的个数

取其中最小的进行枚举（常数优化）

期望得分30分。

1.3 继续优化?

如果要使用在线算法，似乎没有合适的数据结构能回答四维坐标的询问，难以再继续下去。

我们不妨试着考虑离线处理询问

1.4 分块算法

有了离线的想法，再利用分块的思想，不难想到一种复杂度稍优于暴力法的算法。

1.4.1 离散化+离线处理

分别将询问和企鹅按照第一维坐标排序

枚举询问。

1.4.2 对枚举的询问

加入所有第一维比它小的企鹅。

按照第二维排序

每加入sqrt(n)个企鹅，就暴力进行一次排序
其余：暴力。
这部分总复杂度O $(n\sqrt{n})$。

1.4.3 进行一次暴力排序时

已经加入排序的企鹅的个数为M
分块
按照第三维坐标排序
按第四维维护前缀的权值线段树
函数式线段树

1.4.4 询问

整块：二分+查询
非整块：暴力

总复杂度O$(n\sqrt{n}log(n))$
期望得分：60~95

1.5 更优算法

通常，分块处理是在一些复杂度更优的算法存在，但思维复杂度比较高时的一个有效代替品。这里，我们可以稍加思考，找到一个更优的分治算法。

1.5.1

将询问和企鹅一起按第一维坐标排序。
按照第二维坐标进行“归并”。

1.5.2

假设排好序后的编号序列为id[L]~id[R]，其中有的是企鹅，有的是询问，令MID为其终点。

1.5.3

递归处理L~MID，MID+1~R部分的答案。（当然，当L，R相同时就可以直接返回）

1.5.4

我们发现我们还没有处理的情况就只有左半部分（L~MID）的企鹅对右半部分（MID+1~R）的询问的影响，这时左右两部分都已经按第二维坐标排好序(为什么已经排好序见后)，而它们按第一维排的顺序已经不需要了，我们把左半部分和右半部分按第二维坐标进行归并（有序线性表合并)，合并成新的表时，如果添加的是左边的企鹅，就把它三四两维形成的点插入一个“集合”，如果添加的是右边的询问，就在“集合”里询问有多少点在以它三四两维形成的点的左下方，把答案累加到这个询问里。

1.5.5

至于这个“集合”的实现，我们可以用树状数组（或线段树）套平衡树。

总时间复杂度O(nlog^3(n))

1.6 考察点

从在线到离线的思维转换
分块、分治等思想的运用
数据顺序的巧妙处理
数据结构的熟练掌握