Attack 解题报告

杭州学军中学 李超

题目大意：

在平面上有 N 个点，每个点有一个权值，有 M 个操作，每个操作为求一个平行于坐标轴的矩形区域内的点的第 K 小的权值，或交换两个点的权值。N<=30000,M<=30000，每个点的坐标及权值均在 0~10^9 之间。

解题思路：

算法一：

暴力枚举矩形内的点，排序，选出第 K 小的数，暴力修改，时间复杂度 O(NMlogN)，期望得分 40 分。

算法二：

使用 O(N)求第 K 大的算法，暴力修改，时间复杂度 O(NM)，期望得分 50 分。

算法三：

若数据 N 个点在一条直线上，显然这是一个经典的区间第 K 小问题。我们只要对 N 个点构建线段树，每个节点维护它的区间的所有点的平衡树，每次二分答案，将第 K 小问题转为比某个数小的数有多少个，在平衡树中查找并计算，单次的复杂度为 O(log^3N)。对于交换操作，只需在对应的 logN 棵平衡树中修改，单次复杂度为 O(log^2N)，总时间复杂度 O(NlogN+Mlog^3N)。配合算法二，期望得分 60 分。

算法四：

若不存在 SWAP 操作，将所有点按 X 轴坐标构建线段树，每个节点维护以 Y 轴坐标为关键字的线段树，在第二棵线段树的每个节点记录按权值排序队列，对于每个询问，二分答案，同样计算比某个数小的数有多少个，时间复杂度 O(Nlog^2N+Mlog^4N)，配合算法三，期望得分 70 分。

算法五：

在算法四的基础上考虑存在 SWAP 操作的情况，我们发现，只要将维护的队列修改为平衡树，修改时只在对应的树上修改，便可以支持所有情况。时间复杂度仍为 O(Nlog^2N+Mlog^4N)，代码量较大，期望得分与常数关系较大。

算法六：

在一维情况下，如果没有修改操作，我们能够很容易想到划分树这种数据结构，它可以支持查询区间第 K 小的操作，为了支持交换操作，我们可以采用分块的方法，每次交换只修改两个块内的划分树。在二维情况下，我们只要对 X 轴坐标进行分块，块内按 Y 坐标排序，则每个询问在每个块中(除了边界的两个块)都是连续的一段。同样进行二分答案，计算比某个数小的数有多少个。设块大小为 T，则一次查询操作的复杂度为 O(logN*N/T*log(T))，一次交换操作复杂度为 O(TlogT)，当 T=sqrt(NlogN) 时，取到期望最优时间复杂度 O($(N+M)*N^{\frac{1}{2}}*\log N^{\frac{3}{2}}$)，期望得分仍然与常数关系较大。

算法七：

和算法六同样进行分块并构建划分树，我们将每个块中的划分树构成相同的形态，即每棵树的每一层的分割标准均相同，每次询问的时候，由于每棵树的形态相同，可以一起计算出小于等于一个数的数有多少个，然后在所有树中同时向左走或者向右走即可，可以省去二分答案的复杂度。标程使用的方法是离散化后按每个数的二进制表示确定每个数在树中的位置。一次查询操作复杂度为 O(N/T*logN)，总时间复杂度大约为 O($(N+M)*N^{\frac{1}{2}}*\log N$)，期望得分100分。至此，这道题被完美解决了。