我们仍未知道那天所看见的数据结构的名字 解题报告

安徽师范大学附属中学 罗哲正

1 试题来源

IOI2016国家集训队互测第4场

2 试题大意

有一个元素为向量的序列S，下标从0开始，初始时S有一个元素$S_0 = (0, 0)$，现在你需要支持三个操作：

1.在S的末尾添加一个元素(x, y)。

2.删除S的末尾元素。

3.询问下标在$[l, r]$区间内的元素中，$(x, y) \times S_i$的最大值。

其中×表示向量的叉积，$(x_1, y_1) \times (x_2, y_2) = x_1y_2 - x_2y_1$

令n为任意时刻序列长度，m为操作总数，$n \leq 300000; m \leq 500000$；

对于1操作$-10^9 \leq x \leq 10^9; 1 \leq y \leq 10^9$；

对于3操作$1 \leq x \leq 10^9; -10^9 \leq y \leq 10^9$。

3 得分预计

共12人参加考试，预计70分以上2-4人，60分以上3-5人，50分以上4-6人，40分以上5-7人，30分以上6-8人，20分以上10-12人，所有人都能得到至少10分。

4 算法介绍

4.1 Case 1

观察大样例的文件名：unknown1.in/out，猜想这个文件名一定别有用意，结合出题人的良心指数，可以得出结论下发的大样例就是第一个点，于是输出样例即可。

时间复杂度$O(1)$，期望得分0分，实际得分10分。

4.2 Case 2($m \leq 1000$)

对于每个询问，我们暴力枚举$[l, r]$内的所有元素S_i，用$(x, y) \times S_i$更新答案即可。

时间复杂度$O(n^2)$，期望得分20分。

我们观察题目的性质，可以发现我们的询问(x, y)的答案其实是用一条垂直于(x, y)的直线沿着$(-x, -y)$的方向扫过碰到的第一个点。

容易想到这个点一定在所有的S_i构成的凸包上，如果我们能构建出凸包，那么每次只要在凸包上二分斜率，就能做到单词询问$O(\log n)$了。

4.3 Case 3($m \leq 80000$)

对于$m \leq 100000$，我们可以使用分块的做法，把S分成B块，对每块按照x递增的顺序维护一个凸包，插入元素和删除元素都在末尾于是重构最后一个凸包即可，复杂度是$O(\frac{n}{B})$的。

查询时枚举覆盖的整个块，在块内的凸包上二分可以做到每块复杂度$O(\log n)$，对于不覆盖整块的部分，我们使用Case 1-2的方法枚举元素，复杂度是$O(\frac{n}{B})$的。

于是总的复杂度就是$O(B\log n+\frac{n}{B})$，当$B = \sqrt{\frac{n}{\log n}}$时总复杂度最小为$O(n\sqrt{n\log n})$。期望得分30分。

4.4 Case 4($m \leq 300000$，无2操作，3操作询问全部区间)

由于询问都是全局的，我们可以把序列看成一个集合A。

问题就变成了在集合中添加元素，并询问$(x, y) \times A$的最大值，这个问题可以采用二进制分组完成，初始时我们只有一组一个元素，之后每新来一个元素，我们都为它新建一个组，如果有两个组大小相同，我们就把这两个组合并，一个组只需要维护它的凸包，采用归并的方法合并两个组P, Q可以做到合并$O(|P| + |Q|)$。

注意到由于每次合并都是两个相同大小的区间，类似启发式合并，合并总复杂度是$O(n \log n)$的。

这样做对于当前序列长度n，n的每个二进制位都分成了一个组，所以组的个数不会超过$\log n$，对于每个询问，我们枚举每个组并在组内的凸包上二分，每个组内复杂度是$O(n \log^2 n)$。

于是总复杂度就是$O(n \log^2 n)$，期望得分10分。

4.5 Case 5($m \leq 300000$，所有2操作在1操作后面，3操作询问全部区间)

二进制分组只涉及到新建与合并，是不能处理删除操作的，观察这个数据类型的性质，所有的插入都在删除后面，这提示我们可以把操作序列分成两半处理。

于是，对于前一半，Case 4的二进制分组做法，对于后一半，我们把操作倒着做，这样删除操作就变成了插入操作，依旧套用Case 4的做法。

复杂度是$O(n \log^2 n)$的，期望得分20分，双倍送分福利哦。

4.6 Case 6($m \leq 300000$，所有3操作都在1操作和2操作后面)

S序列在询问的时候始终保持不变，这种静态的区间询问问题提示我们使用线段树。

我们对线段树的每个节点建立这个节点上的凸包，建立时直接归并两个孩子的凸包，可以把做到$O(n \log n)$的复杂度。

对于每个询问，在线段树上拆成$O(\log n)$个区间，在每个区间的凸包上二分就行了。

复杂度$O(n \log^2 n)$，期望得分10分。

4.7 Case 7(对于所有3操作有$l = 1$，内存限制为128M)

保证左端点为1。

带操作回溯的题目容易想到建立一棵操作树，我们可以把向量放在边上，这样每个点的序列就是根到这个点的路径。这样每个询问就是树上的一条路径，我们可以把询问放在右端点所在的节点上。

如果询问不是在树上，而是在序列上，由于左端点为1，每次询问区间为一个前缀，那么我们可以采用分治。处理左边对右边的影响时将左边的凸包建出来，右边的所有询问都包含了左边，于是对于右边的每个询问，我们在左边的凸包上查询最优值来更新答案。求凸包可以使用归并算法，复杂度是$O(n \log n)$的。我们同时使用归并将询问按照极角序排序，那么设两个指针分别在询问序列和凸包上扫描，可以把一轮更新做到$O(n)$，于是总复杂度是$O(n \log n)$的。

序列上用分治，树上就用点分治，那么我们把重心的子树拿出来作为右半边，把根到重心的链拿出来作为左半边，就可以用与上面讲的方法更新了。使用两个指针扫描可以做到$O(n \log n)$，或者在凸包上二分的方法可以做到$O(n \log^2 n)$，期望得分30分。

4.8 满分做法

考虑在Case 7做法的基础上，查询的左端点不一定为根。意味着如果我们对根到重心的链一次性建出凸包，不能满足不同左端点的查询要求。怎么办呢？

我们把这个问题拿出来，发现这其实就是在序列上的问题，只是每次询问一个后缀罢了。

于是我们可以以深度为时间轴再进行分治，采用前面讲的归并的方法一次处理做到$O(n \log n)$，于是问题就这么解决了。

时间复杂度$O(n \log^2 n)$，空间复杂度$O(n)$，可以解决全部数据。

4.9 满分做法2

树分治之后，我们统计过重心的询问，以重心为根，那么每个询问就变成了从根到某个节点的链。

然后我们使用splay维护动态凸包，支持加入一个点，撤销最后一次加入，询问和某个向量的点积的最大值。

加入一个点的时候删除的一定是一个区间，我们把这个删除的区间使用splay分割下来并保存起来，撤销的时候只要把这个区间合并回去就可以了。

查询可以简单的通过在Splay上二分做到。

4.10 Case 9(内存限制128M)

按照Case 7做法中的方法建立出操作树，每个询问就是树上的一条路径，我们可以采用树链剖分，将一个树上区间转换成$O(\log n)$个dfs序中的区间。

采用Case 6的做法，对dfs序建立一棵线段树，每次在线段树上查询即可，复杂度是$O(n\log^3 n)$只能得到15分。我们继续优化，我们一个询问用树链剖分结构拆成$O(\log n)$个区间，除了最上面的一个区间之外，每个区间都是这个区间所在重链的前缀。

我们可以对第一个区间在线段树上暴力查询，复杂度是$O(\log^2 n)$。对于其他区间，我们离线处理，对每一条重链自上而下插入节点，采用平衡树维护维护凸包，就可以做到每个查询$O(\log n)$了。

于是总复杂度是$O(n\log^2 n)$，空间复杂度$O(n\log n)$。

4.11 Case 8(内存限制512M)

以上两个算法都需要使用平衡树维护凸包……

什么？你觉得平衡树维护凸包难写？

还有一个不需要点分治也不需要平衡树维护凸包的算法，但是空间复杂度太大了。

Case 9一样，我们把每个查询用树链剖分拆成$O(n\log n)$个区间，然后再用线段树把这$O(n\log n)$个区间拆成$O(n\log^2 n)$个区间。

如果用线段树分治自底而上建立凸包，每次归并两棵子树。对于在线段树某个节点上的询问，我们将询问按照斜率排序，设两个指针在凸包和询问数组上移动，可以在$O(m+q)$时间内完成查询，其中m表示凸包大小，q表示询问个数。

由于线段树区间总长度是$O(n\log n)$，处理在线段树节点上询问的总复杂度为$O(n\log n+Q)$，而$Q=O(n\log^2 n)$。

于是总复杂度就是$O(n\log^2 n)$，空间复杂度是$O(n\log^2 n)$。

4.12 总结&题外话

虽然标题是“我们仍未知道那天所看见的数据结构的名字”，但是我给出的满分算法并没有用到数据结构，而是采用了两次分治简化问题。

其实本质就是把CDQ分治的思想放到树上，序列分治变成点分治，感谢VFleaKing在NOI2014的D2T3首先实现这个思想，并在之后把它分享给我。

P.S.由于~~出题人太过sb~~，以及选手们数据结构水平太高，我每给一个人看着道题，他就能想出来一个新的$O(n\log^2 n)$的数据结构解决方法QAQ，所以就当娱乐题吧……