COT5 解题报告

绍兴一中 张恒捷

1 试题来源

codechef COT5

2 试题大意

在计算机科学中，treap根据键值是一棵二叉搜索树，根据权重是一个堆。
你的任务是维护一棵max-treap（权重大的为根），支持以下操作：
0 k w：插入一个新节点，键值为k，权重为w
1 k：删除键值为k的节点。
2 ku kv：输出键值为ku与kv的节点之间的路径长度。
没两个节点有相同的键值或者权重，并且保证删除时该点存在。
$n \leq 200000$

3 算法介绍

假设u,v的最近公共祖先是w，则

$dist(u,v) = dep(u) + dep(v) - 2 * dep(w)$

键值为 $ku, kv(ku < kv)$ 的点的*lca*为键值在 $[ku, kv]$ 间权重最大的点。这个可以用线段树来维护。

如何计算点的深度

先来考虑*treap*的生成方式：先从所有点中选出权重最大的点作为根，然后将点以键值分成左右两块，分别建出各自的*treap*后作为根的左右儿子。

我们将以键值为关键字排序后的点序列称为 $in-order$ 序列。很显然如果点 u,v 的子树不相交时，必定有一个在 $in-order$ 序列它们之间的权重也比它们大的点。因为这个点会因为权重大成为根，将 u,v 分到不同的子树中。反过来看，在 $in-order$ 序列中，如果点 u,v 间没有比他们权重更大的点，则 u,v 将出现在同一棵子树中，而根是它们中的一个。

所以点 ku 的深度就是满足 kw 权重比 ku 大，且在 $[ku, kw]$ 中没有点权重比 kw 的大的 kw 个数。

我们定义一个序列 A 的上升序列为：$A_i|A_i > A_j(j < i)$。比如序列 $1,4,2,5,6,3$ 的上升序列为 $1,4,5,6$ 。算一个点的深度就相当于算一个序列的上升序列的长度。

考虑使用线段树来维护它：区间 $[l,r]$ 需要维护键值在 $l-r$ 中所有点的权重 max ，与上升序列的长度 len 。另 $calc(l,r,start)$ 为：以 $start$ 为初始最大值，经过这个 $[l,r]$ 区间后上升序列增加的长度。那么可以知道 $len = calc(l,r,0)$ 。

如果 $start > lch.max$ ，那么 $calc(l,r,start) = calc(rch.l, rch.r, start)$

如果 $start \le lch.max$ ，那么 $calc(l,r,start) = calc(lch.l, lch.r, start)+len-lch.len$

这样 $calc$ 就可以用 $O(log\ n)$ 的时间计算了。在执行加入删除操作时只要用线段树，加上时时维护 len 值，就可以在 $log^2 n$ 的时间内解决了。

总时间复杂度： $O(n\ log^2 n)$