T3 Tree Game

假设 Takahashi 一开始选择 r 点为出发点。设 r 点为树根。对于每一个结点 v，我们设一个 $state(v)$： 考虑以 v 为根的子树，假设这两个玩家只能在这棵子树内移动，并且一开始在结点 v。如果先手可以赢得游戏，定义 $state(v) = W$，否则定义它为 L。 可知 $state(r)$ 代表着以r为出发点时的答案。

我们有如下推论： 1.如果结点 v 存在一个子结点 c 满足 $A_c < A_v$ 并且 $state(c) = L$，就可知 $state(v) = W$。 2. 否则，$state(v) = L$。（特别地，如果v是叶节点，$state(v) = L$）

推论 1 的证明：假设轮到你并且当前在结点 v。首先你在 v 拿一个石头（因为 $A_v \neq 0$，所以这是可能的），然后移动到 c。无论对手怎么想从 c 移动到 v 都是不行的，因为你可以从 v 重新移回到 c（注意 $A_v > A_c$，所以可以这么做）。这样，你的对手就只能在以 c 为根的子树中移动了。既然 $state(c) = L$（即c为必败态），你的对手就输了，而你就赢了。（严格地说，你的对手也可以通过在 v 和 c 之间来回移动减少 A_c 的值，但是这并不能改变 c 点的 state）

推论 2 的证明：如果 v 是叶子节点，因为无法从该点移动，因此败下阵来。否则，你可以将当前点从 v 移动到作为 v 子节点之一的 w。这就产生了有两种情况：$state(w) = W$、$A_w \geq A_v$。如果 $state(w) = W$，你的对手就永远无法将棋子移回 v 点，然后在 w 的子树继续继续进行下去。既然 $state(w) = W$，你的对手就赢了，而你输掉了这盘。若 $A_w > A_v$，你的对手一定不会将该棋子通过将其移回 v 的方式，将 u 从 v 移动到 c （这是因为 $A_w > A_v$）。 因此，在这种情况下你同样会输。求取初始点的固定位置，我们可以在 O(N) 求出。此算法的总时间复杂度为 $O(N^2)$。（练习：你可以改进这个算法到 $O(N)$ 吗？）