Lonely King

题目大意

给定一颗 n 个点的叶向树，每个点有权值 c_i，初始每条边均为蓝色。每次操作可以将一条由蓝边形成的路径 $(a_1,a_2),(a_2,a_3),\cdots,(a_k,a_{k+1})$ 替换为红边 (a_1,a_{k+1})。你需要最小化所有可达点对 (u,v) 的权值乘积 $c_u c_v$ 之和。

数据范围

$1 \le n \le 200,000$，$1 \le c_i \le 10^6$。

解题过程

观察最优解的结构：由于替换一条蓝色路径总是会使答案变小，因此最终树上每个叶子会恰好有一条来自祖先的红边。更具体来说，根以外的每棵子树内应该恰有一个叶子经由红边连向子树外，其余的都留在了子树内。

根据上述观察，可以设计如下状态：记 $f_{u,x}$ 表示以 u 为根的子树内，叶子 x 连向子树外时，其余叶子节点所造成贡献的最小值。现在，我们来考虑点 u 恰有两个儿子 v_1,v_2 时的情况，则不难写出转移 $f_{u,y}=f_{v_2,y}+\min_x\left(f_{v_1,x}+c_xc_u\right)$，同理有 $f_{u,x}=f_{v_1,x}+\min_y\left(f_{v_2,y}+c_yc_u\right)$。这是一个标准的一次函数斜截式，即直线 $f(a)=c_xa+f_{v_1,x}$ 和 $g(a)=c_ya+f_{v_2,y}$ 在 c_u 处的最小值，可以使用李超线段树维护，合并两个子树可以使用李超线段树。多个儿子只需按照两个儿子时的情况不断合并即可。

根据李超树合并的复杂度分析，总复杂度为 $\mathcal{O}(n\log n)$。

参考资料

OI Wiki 上关于李超线段树合并的内容。