

MONOPOLY题解

长春吉大附中实验学校 吴一凡

November 3, 2015

1 题目大意

给定一棵有根树，一开始树上的每个节点的颜色均不相同。

从一个点可以走到与这个点相邻的节点，若两个节点颜色不同，则付出1单位的代价；否则不付出代价。

要支持修改操作：给定节点 u ，将从 u 到根节点的路径上所有的节点都染上一种与之前颜色均不相同的新颜色；

还要支持询问操作：给定节点 u ，询问以 u 为根节点的子树中的所有节点到这棵树的根节点的过程中付出的代价的平均值。

数据范围 $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$ 。

2 算法讨论

2.1 算法1

对于修改操作，我们可以暴力 $O(n)$ 处理：先修改颜色，再通过递推求出此时每个点到根节点过程中的代价。

对于询问操作暴力 $O(n)$ 统计即可。

时间复杂度 $O(qn)$ 。

2.2 算法2

要求代价的平均值，我们不妨先求出代价的总和，再除以子树内节点数就能得到平均值。

不妨设以节点 u 为根的子树的节点数为 siz_u 。

当我们要求以 u 为根的子树的答案时，考虑子树内（不包括 u 本身）内的所有节点，如果节点 v 与其父亲的颜色不同，则会提供 siz_v 的贡献（以 v 为根的子树内的节点向上走的过程中都会在这里付出1的代价）；再考虑节点 u 到根节点路径上的所有节点，如果上面某个节点 v 与其父亲的颜色不同，则会提供 siz_u 的贡献。

如果我们能维护每个节点是否产生贡献，就能利用轻重链树链剖分维护链上、子树的贡献之和了。

现在考虑进行一次修改，哪些节点的贡献会发生变化：

设若 u 产生贡献，则 u 的父亲边¹为A类边，否则 u 的父亲边为B类边。

则我们进行修改时，节点 u 到根节点路径上的点的父亲边都会变成B类边，同时节点 u 的所有儿子如果之前是B类边，现在要变成A类边。

这与Link-Cut Tree很相似，A类边就是虚边，B类边就是实边，对 u 进行修改就相当于对 u 在Link-Cut Tree上进行访问。

这样的话，我们可以通过Link-Cut Tree维护树上的每个点是否产生贡献：只需在访问过程中维护每个点的父亲边是实边还是虚边就行了，而访问过程本身就能完成虚实边切换，顺便就能维护。

分析一下复杂度：利用Link-Cut Tree维护树结构复杂度是均摊 $O(\log n)$ 的，于是每次均摊切换 $O(\log n)$ 次虚实边，每次切换都需要用树链剖分维护DFS序上的修改，于是每次均摊复杂度为 $O(\log^2 n)$ 。

总时间复杂度 $O(n \log^2 n)$ 。

¹在有根树意义下，连接节点 u 及其父亲的边