

糖果公园 解题报告

山东省胜利第一中学 王子昱

2013 年 4 月 4 日

Contents

1	题目大意	2
2	数据范围	2
3	算法分析	2
3.1	算法1	2
3.2	算法2	2
3.3	算法3	2
3.3.1	链的情况	3
3.3.2	一般情况	3
3.4	算法4	4
3.5	在线算法	4
4	参考文献	5

1 题目大意

给定一棵 N 个点的树，每个点带权。共有 M 种不同的权值。 Q 组操作，每次可以修改一个点的权值，或是询问连接点 s_i, t_i 的路径的权值。

其中路径的权定义如下：给定数组 V, W ，设路径上第 i 种权值 W_i 的出现次数为 N_w ，则该路径的权值为

$$\sum_i \sum_{j=1}^{N_w} W_i \cdot V_j$$

2 数据范围

对于30%的数据， $N, M, Q \leq 10000$ 。

对于另外20%的数据，没有修改操作，给定的树为链，且 $M \leq 100$ 。

对于另外20%的数据，没有修改操作。

对于另外20%的数据，给定的图为链。

对于所有数据， $N, M, Q \leq 10^5$ 。

3 算法分析

3.1 算法1

对每次询问BFS。时间复杂度 $O((N + M)Q)$ ，期望得分30分。

3.2 算法2

对链上的情况，预处理 $okr[i][j]$ 表示前 i 个点中第 j 种权值的出现次数。利用这一信息，每次询问可以 $O(M)$ 解决。

时间复杂度 $O(M(N + Q))$ ，空间复杂度 $O(MN)$ 。期望得分20分。与算法1结合，可以得到50分。

3.3 算法3

如果没有修改，我们可以用以下方法离线解决。¹

¹由莫涛提出，参见其集训队作业

3.3.1 链的情况

假设树是一条链，将链分为 $N^{\frac{1}{2}}$ 块，将所有询问按照左端点所在块的ID为第一关键字，右端点位置为第二关键字排序后，依次处理每一个询问，动态维护当前询问对应的路径中，每种权值的出现次数。

从询问 (l, r) 转移到询问 (l', r') 的过程中，我们删去仅在前者中出现的点对应的权值，再加上仅在后者中出现的点对应的权值。复杂度显然是左端点的移动距离和右端点移动距离之和。

考虑这一做法的复杂度。对于第一关键字相同的 q 个询问，第一个询问的处理时间为 $O(N)$ ，左端点的移动次数至多为 $O(qN^{\frac{1}{2}})$ ，由于右端点有序，其移动次数为 $O(N)$ 。又因为第一关键字共有 $N^{\frac{1}{2}}$ 种，算法的复杂度为 $O(QN^{\frac{1}{2}})$ 。

3.3.2 一般情况

在树不是链的情况下，将树划分为 $N^{\frac{1}{2}}$ 块，每块的高度不超过 $N^{\frac{1}{2}}$ ，将询问以第一个端点所在的块ID为第一关键字，第二个端点在dfs序列中的位置为第二关键字排序。

考虑从询问 (u_1, v_1) 到 (u_2, v_2) 的转移，设 (u_1, v_1) 之前的询问为 (u_0, v_0) 。采用如下策略：维护以 u_0 和 v_0 为根的树中，每个点的父节点，转移时，将根为 u_0 的树的根转到 u_1 ，将根为 v_0 的树的根转到 v_1 。在根为 u_1 的树中，找出 v_1 和 v_2 的LCA z ，删除结点 $v_1..z$ ，并添加结点 $v_2..z$ 。同样的，在根为 v_1 的树中，找出 u_1 和 u_2 的LCA z' ，删除结点 $u_1..z'$ ，并添加结点 $u_2..z'$ 。

考虑上述过程的复杂度。对 u 在同一个块中的 q 个询问，根从 u_i 换到 u_{i+1} 的复杂度是 $dis(u_i, u_{i+1}) = O(N^{\frac{1}{2}})$ ，从 v_i 换到 v_{i+1} 的总复杂度是

$$\sum dis(v_i, v_{i+1}) = O(N)$$

（由于 v_i 按照dfs序排列），因此，换根操作的总复杂度是 $O(q\sqrt{N} + N)$ 的。对于询问的转移，总复杂度亦为

$$\sum dis(u_i, u_{i+1}) + dis(v_i, v_{i+1}) = O(q\sqrt{N} + N)$$

。因此，算法的总复杂度为 $(N + Q)\sqrt{N}$ 。

利用这一算法可以得到40分。结合算法1可以得到70分。

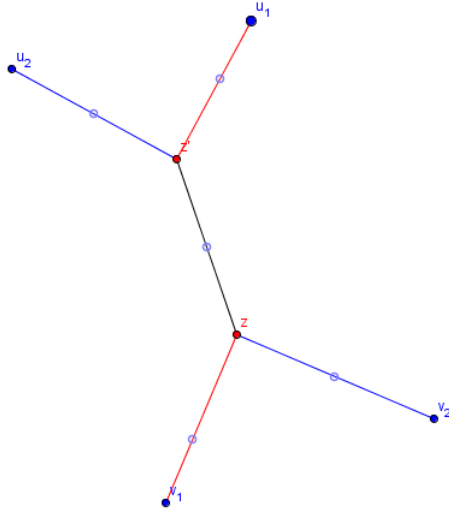


Figure 1: Example of the transition rooted by u_1

3.4 算法4

考虑树是一条链且有修改的情况。我们可以扩展算法3。

将链分为 $N^{1/3}$ 块，将询问按照左端点所在块为第一关键字，右端点所在块为第二关键字，询问之前的修改操作 k 为第三关键字排序。考虑从询问 (l, r, k) 转移到 (l', r', k') 。若 $k > k'$ ，我们撤销第 $k' + 1$ 到第 k 个修改，否则我们增加这些修改。之后端点的转移与算法3同类。

考虑这一做法的复杂度。对于第一、第二关键字相同的 q 个询问，第一个询问的处理时间是 $O(N)$ 的，之后端点转移的代价为 $O(N^{2/3})$ ；一二关键字相同时， k 是单增的， k 的移动次数为 $O(q)$ 。因此，总复杂度为 $O((N + Q)N^{2/3})$ 。

树不是链的情况下，将树按高度分为 $N^{1/3}$ 块，之后的处理方法与上同类，复杂度亦为 $O((N + Q)N^{2/3})$ 。至此，问题完整解决。

3.5 在线算法

事实上，我们有一系列复杂度与上述算法相同的在线解法。详情参见我的集训队论文。

4 参考文献

1. 王子昱,《分块方法的应用》,2013年集训队论文