

dfs序在图论问题的应用

赵涵铮

2023 年 2 月 18 日

1 异象石

1.1 题目描述

给定有 n 个节点的树， m 次操作，每次操作：

1. 选定一个点作为异象点
2. 删除一个异象点
3. 输出联通所有异象点的路径之和

1.2 数据范围

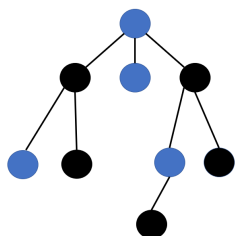
$1 \leq N \leq 10^5, 1 \leq M \leq 10^5, 1 \leq X \leq Y \leq N$, 数字不超过 C/C++ 的 int 范围。

1.3 解题思路

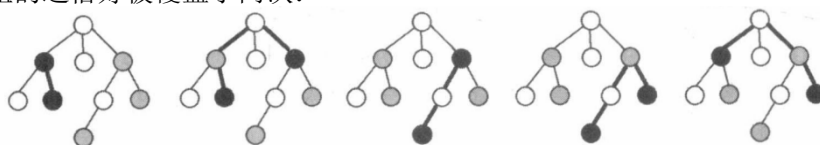
根据题意，本题是对一棵树进行维护，由于树上两点只有唯一路径，所以是不需要考虑最短路径等问题的，只需要考虑如何维护临近的两点。

我们可以先通过dfs，得到每一个点的时间戳，然后按照时间戳顺序，将异象点的节点排成首尾相连的一圈，累加两点之间的路径长度，最终得到的结果恰好是答案的两倍（每条边恰好经过两次）

以此图为例，假设黑色的点为异象点：



按照时间戳顺序，以此选择两个点，加粗表示两者路径，通过五次计算，加粗的边恰好被覆盖了两次：



为了维护好这个异象点集合，我们可以使用数据结构`set`，便于增删点，用`ans`来记录序列相邻点的路径长度之和（包括序列首尾）。

接下来，我们需要解决如何计算任意两点距离的问题。设 $path(x, y)$ 表示树上两点的路径长度，设 $d[x]$ 表示 x 到根节点的路径长度，可知：

$$path(x, y) = d[x] + d[y] - 2 * d[LCA(x, y)] \quad (1)$$

这样，就可以通过 LCA 算出 $path(x, y)$ ，同时，我们可以通过一次 dfs 预处理好 d 数组。

现在，让我们再次回顾一下全过程：若一个节点 x 出现了异象石，则根据时间戳，将节点 x 插入`set`中，它前后分别是 l, r ，则

$$ans = path(l, r) + path(l, x) + path(x, r)$$

若一个节点的异象点被删除，就类似的更新`set`。值得一提的是，`set`的插入和删除复杂度都是 $O(\log N)$ ，时间复杂度为 $O((N + M) \log N)$

1.4 参考代码