

Query on a tree VI 解题报告

吉大附中实验学校 王天懿

October 28, 2015

1 题目描述

给定一棵 n 个节点的树，每个节点有一个颜色(黑/白)，初始都为黑。

维护一种数据结构，支持下列操作：

-询问有多少个点与给定的点 u 连通。两个点 u 和 v 是连通的，当且仅当 u 到 v 最短路径上的所有点(包括 u 和 v)颜色都相同。

-切换一个点的颜色(黑变白，白变黑)。

2 数据规模与约定

$1 \leq n, m \leq 10^5$

其中 m 为操作次数。

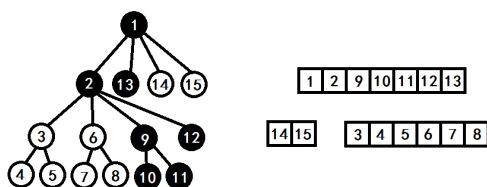
3 题目解法

我使用了Splay维护DFS序的做法来解决这个问题。

为了方便称呼，我们将彼此连通的极大点集称作一个连通块。

一个朴素的想法是，我们将每个连通块都用一棵Splay来维护它的DFS序。但是这样会导致一个问题，就是我们在切换一个点颜色时所进行的操作次数是与这个点的度数相关的，而一个点的度数可以达到 $O(n)$ 级别，如果反复对一个度数非常大的点进行切换颜色操作，就可能导致代码超时。

因此，我们不妨考虑将同一节点的所有异色子连通块¹合并到同一棵Splay中，如下图所示：



注意图中的以3号节点为根的子树和以6号节点为根的子树，虽然这两棵子树分属于不同的连通块，但是由于它们都是2号节点的异色子连通块，因此被放在了同一棵Splay中。14和15两个节点同理。

接下来我们考虑切换颜色和询问所在连通块大小这两个操作如何实现。

3.1 切换节点 u 的颜色

我们用一个指针记录一个节点的所有异色子连通块所在Splay的根节点。

首先，我们将点 u 所在连通块中 u 节点子树的子树从所在Splay中取出。显然这些节点在DFS序中都是排在一起的。

接下来，考虑到原先的同色子连通块变成了异色子连通块，异色子连通块变成了同色子连通块，因此我们可以考虑交换这两部分。异色子连通块已经用指针记录了，同色子连通块即我们刚取出的序列中除掉点 u 外的剩

¹这里不妨将一个点的子节点在连通块中子树称作子连通块。

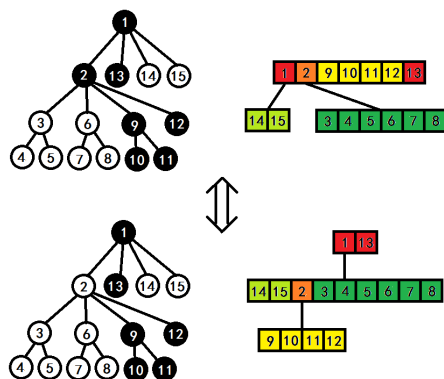
余部分。

现在我们考虑节点 u 和 $fa[u]$ 之间的关系。

如果切换颜色后，节点 u 和 $fa[u]$ 颜色相同，那么点 u 所在连通块成为了 $fa[u]$ 的同色子连通块，将点 u 所在Splay插入 $fa[u]$ 在Splay中对应节点的后面即可。

否则，将点 u 所在Splay插入 $fa[u]$ 的异色子连通块所在Splay的最后即可。

具体操作如图所示：



将点2的颜色由黑色切换为白色时，首先将点2在连通块中的子树(2-9-10-11-12)取出，然后交换同色子连通块(9-10-11-12)和异色子连通块(3-4-5-6-7-8)，最后将点2在连通块中的子树(2-3-4-5-6-7-8)插入点1的异色子连通块(14-15)的最后即可。

将点2的颜色由白色切换为黑色时，前一部分操作与上述相同，不同的是最后要将点2在连通块中的子树(2-9-10-11-12)插入点1的后面。

3.2 询问节点 u 所在连通块的大小

维护了上述数据结构后，询问就显得简单很多了。

只需要找到点 u 所在连通块的根节点 r ，找到 r 在连通块中的子树并输出其 $size$ 即可。

那么如何找到 r 呢？

首先我们找到 u 所在Splay的根 r' ，那么显然有 $fa[r'] = fa[r]$ 。

现在我们知道了 $fa[r]$ ，我们想要找到 r ，这个可以利用树上倍增算法在 $O(\log n)$ 的时间内完成。

3.3 找到节点 u 在连通块中的子树对应的Splay

这个问题等价于在节点 u 所在Splay中找到最后面的一个节点 v ，满足 $LCA(u, v) = u$ 。

在Splay上二分查找，利用RMQLCA算法可以 $O(n \log n)$ 预处理， $O(1)$ 查询两个点的LCA。

为了保证时间复杂度，找到节点 v 后还要把 v 旋转到Splay的根上。

4 时空复杂度

时间复杂度： $O((n + m) \log n)$

空间复杂度： $O(n \log n)$

5 其他做法

其实只要本着“异色子连通块放在一起维护”这个思想来做这道题的话，许多常见的树上数据结构(树链剖分，Link Cut Tree，Top Tree等)都是可以解决这道题的。

我使用Splay维护DFS序是因为这个算法相对便于理解，且不需要信息可减(可以直接拓展到询问连通块最小权值的Query On A Tree VII上)，作为代价来说实现上面略显繁琐，常数稍大。