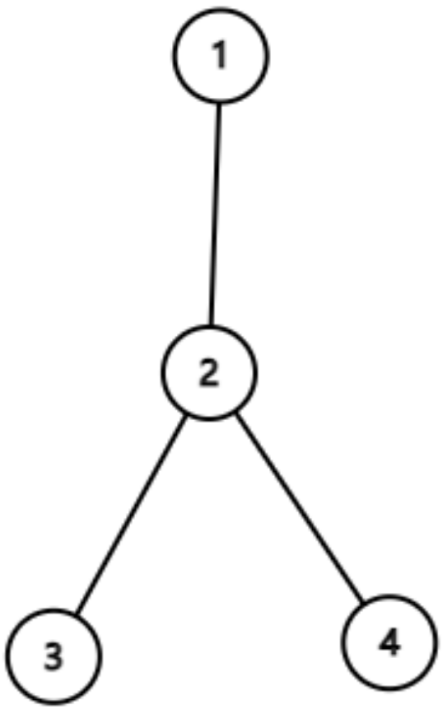
最近公共祖先

LCA

stff577

2023 年 1 月 12 日

概念



在有根树中，祖先指从根到该节点所经分支上的所有节点。在这里，一个节点也可以是它自己的祖先。$LCA(1,2)$ 为 2，$LCA(3,4)$ 为 2

求解方法

- 考虑最暴力的做法

求解方法

- 考虑最暴力的做法
- 因为在有根树上，除了根节点，我们能确定每个节点有且仅有一个父节点，所以先标记一个节点的所有祖先节点，然后再从另一个节点往父节点遍历，遇到的第一个被标记过的节点就是最近公共祖先了

求解方法

- 考虑最暴力的做法
- 因为在有根树上，除了根节点，我们能确定每个节点有且仅有一个父节点，所以先标记一个节点的所有祖先节点，然后再从另一个节点往父节点遍历，遇到的第一个被标记过的节点就是最近公共祖先了
- 一次询问的时间复杂度为 $O(n)$

问题引入

- 给定一棵有根树，树上有一些特殊节点，每次询问给一个点，问它的祖先节点中，距离它最近的特殊节点是哪个
- $n, m, q \leq 10^5$

离线 tarjan

- 先把所有询问读进来，每个节点开一个 $vector < int >$，记录这是第几个询问。同时开一个 ans 数组，$ans[i]$ 代表第 i 次询问的答案

离线 tarjan

- 先把所有询问读进来，每个节点开一个 $vector < int >$，记录这是第几个询问。同时开一个 ans 数组，$ans[i]$ 代表第 i 次询问的答案
- 从根节点进行一次 dfs，设一个变量，遇到特殊节点之后就将变量的值改成当前最近遇到的特殊节点，dfs 出来时再变回原来的版本

离线 tarjan

- 先把所有询问读进来，每个节点开一个 $vector < int >$，记录这是第几个询问。同时开一个 ans 数组，$ans[i]$ 代表第 i 次询问的答案
- 从根节点进行一次 dfs，设一个变量，遇到特殊节点之后就将变量的值改成当前最近遇到的特殊节点，dfs 出来时再变回原来的版本
- 遍历每个节点的 $vector$，将最近一次遇到的特殊节点的编号赋值给对应的 $ans[i]$

离线 tarjan

- 先把所有询问读进来，每个节点开一个 $vector < int >$，记录这是第几个询问。同时开一个 ans 数组，$ans[i]$ 代表第 i 次询问的答案
- 从根节点进行一次 dfs，设一个变量，遇到特殊节点之后就将变量的值改成当前最近遇到的特殊节点，dfs 出来时再变回原来的版本
- 遍历每个节点的 $vector$，将最近一次遇到的特殊节点的编号赋值给对应的 $ans[i]$
- 遍历 ans 数组进行输出

离线 tarjan

- 先把所有询问读进来，每个节点开一个 $vector < int >$，记录这是第几个询问。同时开一个 ans 数组，$ans[i]$ 代表第 i 次询问的答案
- 从根节点进行一次 dfs，设一个变量，遇到特殊节点之后就将变量的值改成当前最近遇到的特殊节点，dfs 出来时再变回原来的版本
- 遍历每个节点的 $vector$，将最近一次遇到的特殊节点的编号赋值给对应的 $ans[i]$
- 遍历 ans 数组进行输出
- 时间复杂度 $O(n+q)$

离线 tarjan

- 受上述算法启发，我们是不是也能先按照本身的性质，对询问进行回答，最后按序输出呢

离线 tarjan

- 受上述算法启发，我们是不是也能先按照本身的性质，对询问进行回答，最后按序输出呢
- 我们对 LCA 进行一个变换，两点的 LCA 也可以看成找一个深度最大的节点，使得询问的两点均在以该节点为根的子树中

离线 tarjan

- 受上述算法启发，我们是不是也能先按照本身的性质，对询问进行回答，最后按序输出呢
- 我们对 LCA 进行一个变换，两点的 LCA 也可以看成找一个深度最大的节点，使得询问的两点均在以该节点为根的子树中
- 我们对 LCA 进行分类讨论，即 LCA 为其中一节点的情况（换句话说，其中一个节点为另一个节点的祖先节点），或 LCA 为另外一节点的情况

离线 tarjan

- 受上述算法启发，我们是不是也能先按照本身的性质，对询问进行回答，最后按序输出呢
- 我们对 LCA 进行一个变换，两点的 LCA 也可以看成找一个深度最大的节点，使得询问的两点均在以该节点为根的子树中
- 我们对 LCA 进行分类讨论，即 LCA 为其中一节点的情况（换句话说，其中一个节点为另一个节点的祖先节点），或 LCA 为另外一节点的情况
- 不知道会先遍历到询问的哪个点，所以询问要挂在两个点上。对于第一种情况，只需要标记另外一个节点是否已经被遍历过就行了

离线 tarjan

- 受上述算法启发，我们是不是也能先按照本身的性质，对询问进行回答，最后按序输出呢
- 我们对 LCA 进行一个变换，两点的 LCA 也可以看成找一个深度最大的节点，使得询问的两点均在以该节点为根的子树中
- 我们对 LCA 进行分类讨论，即 LCA 为其中一节点的情况（换句话说，其中一个节点为另一个节点的祖先节点），或 LCA 为另外一节点的情况
- 不知道会先遍历到询问的哪个点，所以询问要挂在两个点上。对于第一种情况，只需要标记另外一个节点是否已经被遍历过就行了
- 对于第二种情况，也要对遍历过的节点打标记，同时发现需要把先遍历过的点往上提，提到最近一个还没有退出 dfs 的节点上来。这里用并查集实现就很方便了，注意合并方向

离线 tarjan

- 受上述算法启发，我们是不是也能先按照本身的性质，对询问进行回答，最后按序输出呢
- 我们对 LCA 进行一个变换，两点的 LCA 也可以看成找一个深度最大的节点，使得询问的两点均在以该节点为根的子树中
- 我们对 LCA 进行分类讨论，即 LCA 为其中一节点的情况（换句话说，其中一个节点为另一个节点的祖先节点），或 LCA 为另外一节点的情况
- 不知道会先遍历到询问的哪个点，所以询问要挂在两个点上。对于第一种情况，只需要标记另外一个节点是否已经被遍历过就行了
- 对于第二种情况，也要对遍历过的节点打标记，同时发现需要把先遍历过的点往上提，提到最近一个还没有退出 dfs 的节点上来。这里用并查集实现就很方便了，注意合并方向
- 时间复杂度 $O(n + q \cdot \alpha(n))$

在线倍增

- 如果需要按照顺序来回答，我们可以采用倍增的方法来实现

在线倍增

- 如果需要按照顺序来回答，我们可以采用倍增的方法来实现
- 对于 LCA 为其中一点的情况，只需要用深度较大的节点往上跳，找到其祖先节点中，深度最小的节点，满足其深度不小于询问的另外一个节点。此时答案就是该节点

在线倍增

- 如果需要按照顺序来回答，我们可以采用倍增的方法来实现
- 对于 LCA 为其中一点的情况，只需要用深度较大的节点往上跳，找到其祖先节点中，深度最小的节点，满足其深度不小于询问的另外一个节点。此时答案就是该节点
- 对于 LCA 为另外一节点的情况，我们参考第一种情况的处理方式，此时两个节点的深度相同，我们需要两个节点同时往上跳，找到第一次相遇的节点。但倍增只能找到符合或者不符合条件的最后一个节点，所以我们在跳的过程中找最后一对不相等的节点，该节点的父亲节点就是我们要的 LCA

在线倍增

- 如果需要按照顺序来回答，我们可以采用倍增的方法来实现
- 对于 LCA 为其中一点的情况，只需要用深度较大的节点往上跳，找到其祖先节点中，深度最小的节点，满足其深度不小于询问的另外一个节点。此时答案就是该节点
- 对于 LCA 为另外一节点的情况，我们参考第一种情况的处理方式，此时两个节点的深度相同，我们需要两个节点同时往上跳，找到第一次相遇的节点。但倍增只能找到符合或者不符合条件的最后一个节点，所以我们在跳的过程中找最后一对不相等的节点，该节点的父亲节点就是我们要的 LCA
- 时间复杂度 $O((n+q)\cdot log(n))$

在线倍增

- 为什么不是二分？

在线倍增

- 为什么不是二分？
- 有根树中节点至多只有一个确定的父节点，但是可能多有个子节点，无法直接通过下标顺序确定该节点后面第 x 个节点是谁

在线倍增

- 为什么不是二分？
- 有根树中节点至多只有一个确定的父节点，但是可能多有个子节点，无法直接通过下标顺序确定该节点后面第 x 个节点是谁
- 如果从子节点往上二分，由于每个子节点对应的祖先各不相同，用线性表的话需要独立存储

求树上两点距离

- 设 $dis[u]$ 为点 u 到根节点的距离（在无权树当中，可以直接用深度来代替），点 u 到点 v 的树上距离即为 $dis[u] + dis[v]$ - $2 * dis[lca(u, v)]$

例题代码

- 离线 *tarjan*：https://www.luogu.com.cn/paste/x0nicuro
- 在线倍增：https://www.luogu.com.cn/paste/4f4n45nu

谢谢！