# [二叉树的最近公共祖先](https://leetcode-cn.com/explore/interview/card/bytedance/244/linked-list-and-tree/1026/)

## 题目

给定一个二叉树, 找到该树中两个指定节点的最近公共祖先。

百度百科中最近公共祖先的定义为：“对于有根树 T 的两个结点 p、q，最近公共祖先表示为一个结点 x，满足 x 是 p、q 的祖先且 x 的深度尽可能大（一个节点也可以是它自己的祖先）。”

示例 1:

输入: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 1
输出: 3
解释: 节点 5 和节点 1 的最近公共祖先是节点 3。

## 解题思路

1. 通过 DFS 找到节点的路径
2. 从头开始遍历两个节点的路径，找到最后一个相等的节点

public TreeNode lowestCommonAncestor(TreeNode root, TreeNode p, TreeNode q) {
 LinkedList<TreeNode> pathP = new LinkedList<>();
 LinkedList<TreeNode> pathQ = new LinkedList<>();

 findNodePath(pathP, root, p);
 findNodePath(pathQ, root, q);

 TreeNode last = null;
 while (!pathP.isEmpty() && !pathQ.isEmpty()) {
 TreeNode pi = pathP.pollFirst();
 TreeNode qi = pathQ.pollFirst();

 if (qi==pi) {
 last = pi;
 }else break;

 }

 return last;
}

private void findNodePath(LinkedList<TreeNode> path, TreeNode root, TreeNode target) {
 if (root == null) {
 return;
 }

 if (!path.isEmpty() && path.getLast().val == target.val) {
 return;
 }

 path.addLast(root);

 findNodePath(path, root.left, target);
 if (!path.isEmpty() && path.getLast().val == target.val) {
 return;
 }

 findNodePath(path, root.right, target);

 if (!path.isEmpty() && path.getLast().val != target.val) {
 path.removeLast();
 }
}