# 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。

**示例 2:**

****

输入: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 4

输出: 5

解释: 节点 5 和节点 4 的最近公共祖先是节点 5。因为根据定义最近公共祖先节点可以为节点本身。

**提示：**

树中节点数目在范围 [2, 105] 内。

-109 <= Node.val <= 109

所有Node.val互不相同。

p != q

p和q均存在于给定的二叉树中。

注：参考Leetcode 235

# 分析

## 方法一：递归法

class Solution {

public:

TreeNode\* lowestCommonAncestor(TreeNode\* root, TreeNode\* p, TreeNode\* q) {

if(root == nullptr || root == p || root == q) return root;

TreeNode \*left = lowestCommonAncestor(root->left, p, q);

TreeNode \*right = lowestCommonAncestor(root->right, p, q);

if(left == nullptr) return right;

if(right == nullptr) return left;

return root;

}

};

## 方法二：迭代法