

查找两个二叉树两个节点的公共祖先

任何递归型的问题，无非就是灵魂三问：1、这个函数是干嘛的？2、这个函数参数中的变量是什么的？3、得到函数的递归结果，你应该干什么？

解题思路：描述：给定三个节点`root`，`p`，`q`，返回一个节点。情况1：如果`p,q`都在以`root`为根的二叉树中，直接返回`root`；情况2：如果`p,q`都不在以`root`为根的二叉树中，直接返回`nullptr`；情况3：如果`p,q`只有一个存在在以`root`为根的二叉树种，返回该节点。

```
class Solution {
public:
    TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q)
    {
        if (root == nullptr) {
            return nullptr;
        }

        if (root == p || root == q) {
            return root;
        }

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

        // 情况1
        if (left && right) {
            return root;
        }

        // 情况2
        if (!left && !right) {
            return nullptr;
        }

        // 情况3
        return (left == nullptr ? right : left);
    }
};
```