

236. 二叉树的最近公共祖先

题目大意

求二叉树最近公共祖先。

为了结构，虽然只有一句话：-(

题目分析

树上节点数目 $n \leq 1e5$ 。题目只询问一次，没有必要用算法进行预处理了，我们观察到一共有两种情况：

1. 询问两节点在一条链上，此时 LCA 为深度较浅的节点
2. 不在一条链上，LCA 的左子树中存在目标节点，且右子树中也存在目标节点

不难发现，满足这样条件的节点是唯一的，因此只要搜到了这样的节点，就是 LCA。因此，我们只需要对树进行搜索，然后对“左子树中是否出现”，“右子树中是否出现”以及“自身是否是”三个条件的满足情况进行计数，若计数结果为 2，则其为题目所要求的 LCA，时间复杂度为 $O(n)$ 。代码如下：

```
TreeNode *tar1, *tar2, *ret = nullptr;
bool find(TreeNode* now){
    int found = 0;
    if(now->left != nullptr) found += find(now->left);
    if(now->right != nullptr) found += find(now->right);
    if(now == tar1 or now == tar2) found++;
    if(found == 2 and ret==nullptr) ret = now;
    return found;
}
TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {
    tar1 = p, tar2 = q;
    find(root);
    return ret;
}
```