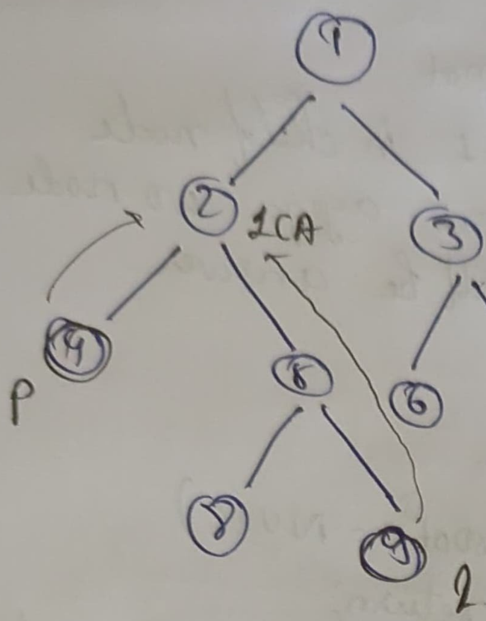
LCA (Lowest Common Ancestor) {LC:236}

Note:- A node can also be ancestor to itself

node → p, q (p != q)



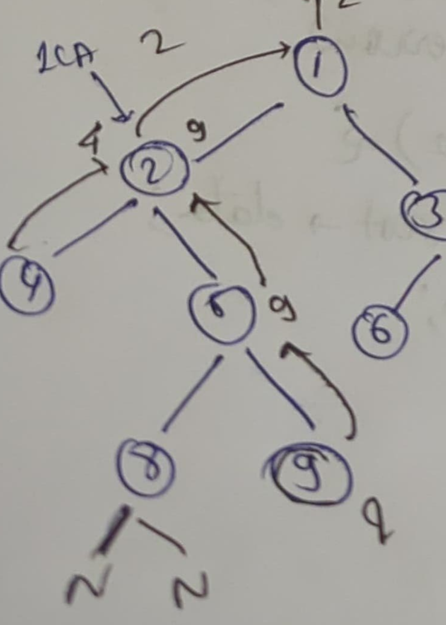
example { if p = 5, q = 7
LCA = 1

{ if p = 6, q = 7
LCA = 3

{ if p = 6, q = 3
LCA = 3

*** { LCA
↓
1st node of subtree in which p & q exists

Approach: check if p & q exist in left or right subtree



4 Cases:

① left LCA = NULL && right LCA = NULL
return NULL

②&③ left LCA valid || right LCA valid
return valid value

④ ! left LCA & ! right LCA ⇒ both not null
return root (dono valid value)
↳ LCA

Pseudo Code

```
Node* LCA(root, p, q){
      if(root == NULL) return NULL;
    if (root == p || root == q)
          return root;
    leftLCA = LCA (root→left, p, q)
    rightLCA = LCA ( root → right, p, q)
    if (leftLCA && rightLCA)
           return root;
    else if (leftLCA != NULL)
           return leftLCA;
      else
           return rightLCA;
}
```

236. Lowest Common Ancestor of a Binary Tree

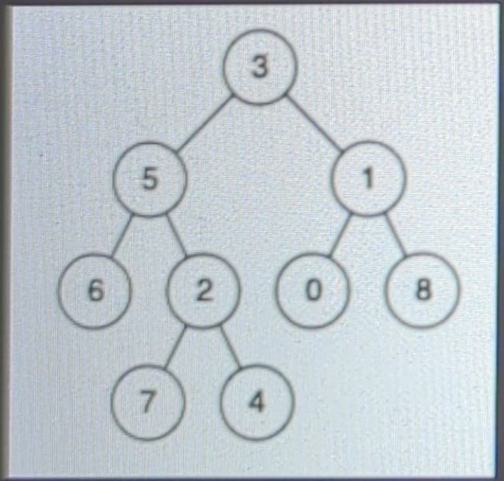
Solved ✓

Medium Topics Companies

Given a binary tree, find the lowest common ancestor (LCA) of two given nodes in the tree.

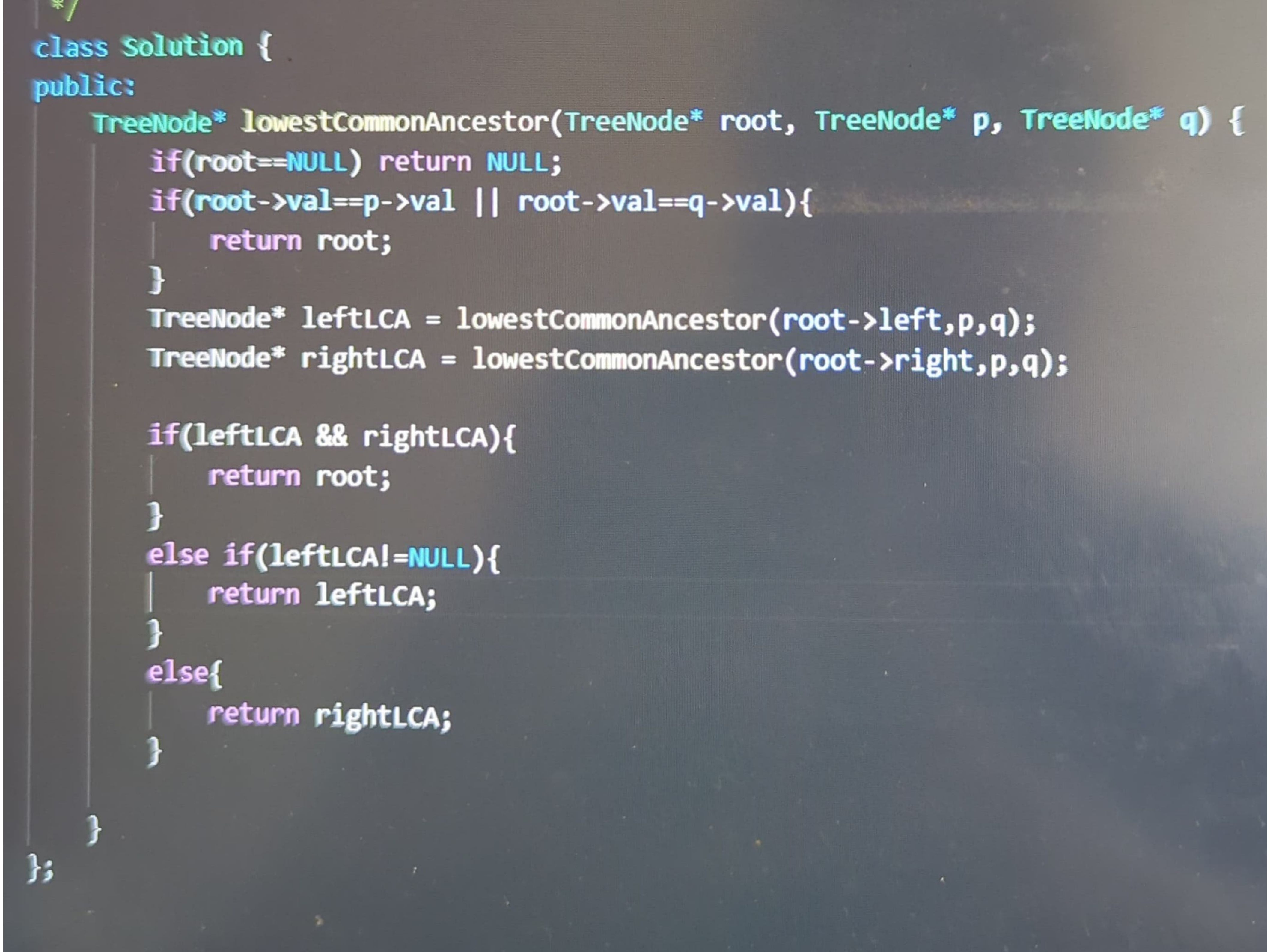
According to the definition of LCA on Wikipedia: "The lowest common ancestor is defined between two nodes `p` and `q` as the lowest node in `T` that has both `p` and `q` as descendants (where we allow **a node to be a descendant of itself**)."

Example 1:



```
Input: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 1
Output: 3
Explanation: The LCA of nodes 5 and 1 is 3.
```

Example 2:



```
*/
class Solution {
public:
    TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {
        if(root==NULL) return NULL;
        if(root->val==p->val || root->val==q->val){
            return root;
        }
        TreeNode* leftLCA = lowestCommonAncestor(root->left,p,q);
        TreeNode* rightLCA = lowestCommonAncestor(root->right,p,q);

        if(leftLCA && rightLCA){
            return root;
        }
        else if(leftLCA!=NULL){
            return leftLCA;
        }
        else{
            return rightLCA;
        }

    }
};
```