236. Lowest Common Ancestor of a Binary Tree

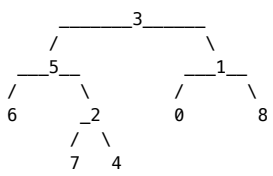
Add to List

Question | Editorial Solution | My Submissions (/problems/lowest-common-ancestor-of-a-binary-tree/submissions/)

Total Accepted: **81082** Total Submissions: **276348** Difficulty: **Medium** Contributors: **Admin**

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 (https://en.wikipedia.org/wiki/Lowest_common_ancestor): “The lowest common ancestor is defined between two nodes v and w as the lowest node in T that has both v and w as descendants (where we allow **a node to be a descendant of itself**).”



For example, the lowest common ancestor (LCA) of nodes `5` and `1` is `3`. Another example is LCA of nodes `5` and `4` is `5`, since a node can be a descendant of itself according to the LCA definition.

Hide Company Tags | Amazon (/company/amazon/) | LinkedIn (/company/linkedin/) | Apple (/company/apple/) | Facebook (/company/facebook/) | Microsoft (/company/microsoft/)

Hide Tags | Tree (/tag/tree/)

Hide Similar Problems | (E) Lowest Common Ancestor of a Binary Search Tree (/problems/lowest-common-ancestor-of-a-binary-search-tree/)

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/292) | Top Solutions | Pick One (/problems/random-one-question/)

C++

```cpp
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {
        if(root==NULL||root==p||root==q) return root;
        TreeNode* left = lowestCommonAncestor(root->left,p,q);
        TreeNode* right = lowestCommonAncestor(root->right,p,q);
        if(left&&right) return root;
        return (left!=NULL)?left:right;
    }

};
```

Submit Solution