235. Lowest Common Ancestor of a Binary Search Tree

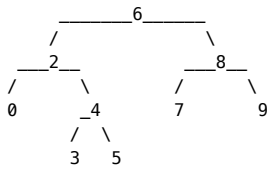
Add to List

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

Total Accepted: **116088** Total Submissions: **304225** Difficulty: **Easy** Contributors: **Admin**

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

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 `2` and `8` is `6`. Another example is LCA of nodes `2` and `4` is `2`, since a node can be a descendant of itself according to the LCA definition.

Hide Company Tags | Amazon (/company/amazon/) | Microsoft (/company/microsoft/) | Facebook (/company/facebook/) | Twitter (/company/twitter/)

Hide Tags | Tree (/tag/tree/)

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

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

Discuss (https://discuss.leetcode.com/category/291) | Top Solutions | Pick One (/problems/random-one-question/)

Notes

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(p==NULL) return q;
        if(q==NULL) return p;
        while(root!=NULL&&root!=p&&root!=q){
            if(root->val>p->val&&root->val>q->val) root = root->left;
            else if(root->val<p->val&&root->val<q->val) root = root->right;
            else break;
        }
        return root;
    }
};
```

Submit Solution