285. Inorder Successor in BST

Add to List

Question | Editorial Solution | My Submissions (/problems/inorder-successor-in-bst/submissions/)

Total Accepted: **24054** Total Submissions: **66522** Difficulty: **Medium** Contributors: **Admin**

Given a binary search tree and a node in it, find the in-order successor of that node in the BST.

Note: If the given node has no in-order successor in the tree, return `null`.

Notes

Hide Company Tags | Pocket Gems (/company/pocket-gems/) | Microsoft (/company/microsoft/) | Facebook (/company/facebook/)

Hide Tags | Tree (/tag/tree/)

Hide Similar Problems | (M) Binary Tree Inorder Traversal (/problems/binary-tree-inorder-traversal/) | (M) Binary Search Tree Iterator (/problems/binary-search-tree-iterator/)

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

Discuss (https://discuss.leetcode.com/category/355) | 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) {}
 * };
 */

 //works for general Binary Tree not restricted to BST
class Solution {
public:
    TreeNode* inorderSuccessor(TreeNode* root, TreeNode* p) {
        if(p==NULL) return NULL;
        if(p->right!=NULL) {
            TreeNode* tmp = p->right;
            while(tmp->left!=NULL) tmp=tmp->left;
            return tmp;
        }
        TreeNode* res = helper(root,p);
        if(res==p)
            return NULL;
        else
            return res;
    }
private:
    TreeNode* helper(TreeNode* root, TreeNode* p){
        if(root==NULL) return NULL;
        if(root==p) return p;

        TreeNode* left = helper(root->left,p);
        if(left==p) return root;
        if(left!=NULL) return left;
        TreeNode* right = helper(root->right,p);
        if(right==p) return p;
        if(right!=NULL) return right;
        return NULL;
    }
};
```

Submit Solution