Problem

Editorial

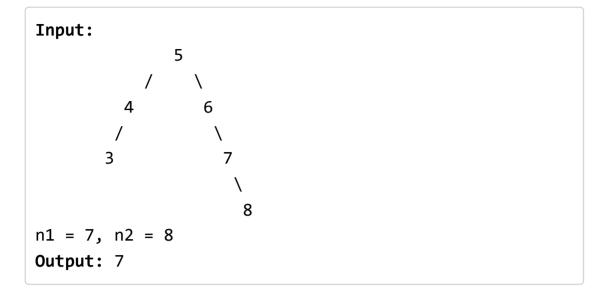
Submissions

Doubt Support

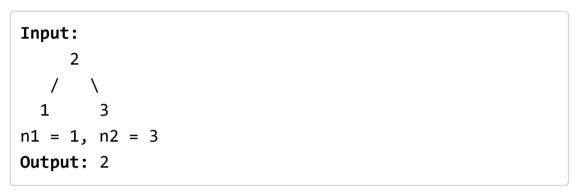
Lowest Common Ancestor in a BST \square Easy Accuracy: 50.22% Submissions: 86135 Points: 2

Given a Binary Search Tree (with all values unique) and two node values. Find the Lowest Common Ancestors of the two nodes in the BST.

Example 1:



Example 2:



Your Task:

You don't need to read input or print anything. Your task is to complete the function LCA() which takes the root Node of the BST and two integer values n1 and n2 as inputs and returns the Lowest Common Ancestor of the Nodes with values n1 and n2 in the given BST.

Expected Time Complexity: O(Height of the BST). **Expected Auxiliary Space:** O(Height of the BST).

Constraints:

ปนุเษย เลายน

View Bookmarked Problems (https://practice.geeksforgeeks.org/explore/? problemType=bookmark)

Problem Solved Successfully



You get marks only for the first correct submission if you solve the problem without **volus** viewing the full solution.

Related Courses lest Cases Passed:

Total Time Taken:

Red Interview Experiences

0.2/1.38

We are replacing the old Disqus forum with the new Discussions section given below. Click here (https://practice.geeksforgeeks.org/comments/lowest-common-ancestor-ina-bst/1/?rel=https://practice.geeksforgeeks.org/problems/lowest-common-ancestor-Correct Submission Counts: 1) to view old DA the month lefts.

8 6

Report An Issue

If you are facing any issue on this page. Please let us know.

```
C++ (g++ 5.4)
                  Test against custom input
103 /*
104
        LCA = Stores node pointer
105
         ans = \&LCA;
106
107
         ans means address of LCA
108
         *ans means value of LCA(LCA value is nothing but the address of Node)
109
         **ans means value pointed by
110 */
111
    void solve(Node* root, int n1, int n2, Node** ans){
         if(root == NULL)
112
113
             return;
114
115
         if((root->data) < n1 and (root->data) < n2)</pre>
116
             solve(root->right, n1, n2, ans);
117
         else if((root->data) > n1 and (root->data) > n2)
118
             solve(root->left, n1, n2, ans);
119
         else *ans = root;
120 }
121 Node* LCA(Node *root, int n1, int n2)
122 {
123
        //Your code here
124
125
```

<u>-ÿ</u>.