



Lowest Common Ancestor in a BST



Easy

Accuracy: 65.2%

Submissions: 106K+

Points: 2

Explore Job Fair for students & freshers for daily new opportunities.

Find A Job Today!



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:

Input:



n1 = 7, n2 = 8

Output: 7

```
1 // } Driver Code Ends
99
100
101 //Function to find the lowest common ancestor in a BST.
102 Node* LCA(Node *root, int n1, int n2)
103 {
104     if(root==NULL){
105         return NULL;
106     }
107     if(root->data < n1 && root->data < n2){
108         return LCA(root->right,n1,n2);
109     }
110     if(root->data > n1 && root->data > n2){
111         return LCA(root->left,n1,n2);
112     }
113     return root;
114 }
115
116
117
118
```



Custom Input

Compile & Run

Submit





Find a pair with given target in BST



Medium

Accuracy: 44.02%

Submissions: 44K+

Points: 4

Explore Job Fair for students & freshers for daily new opportunities.

Find A Job Today!



Given a Binary Search Tree and a target sum. Check whether there's a pair of Nodes in the BST with value summing up to the target sum.

Example 1:

Input:

```
      2
     / \
    1   3
```

sum = 5

Output: 1

Explanation:

Nodes with value 2 and 3 sum up to 5.

Example 2:

```
86 struct Node {
87     int data;
88     Node *left;
89     Node *right;
90
91     Node(int val) {
92         data = val;
93         left = right = NULL;
94     }
95 };
96
97
98 class Solution{
99 public:
100     // root : the root Node of the given BST
101     // target : the target sum
102     bool inorder(Node *root,int target,unordered_set<int>&s){
103         if(root==NULL){
104             return false;
105         }
106         if(inorder(root->left,target,s)==true){
107             return true;
108         }
109         if(s.find(target-root->data)!=s.end()){
110             return true;
111         }
112         s.insert(root->data);
113         return inorder(root->right,target,s);
114     }
115
116     int isPairPresent(struct Node *root, int target)
117     {
118         unordered_set<int>st;
119         return inorder(root,target,st);
120     }
121 };
122 // } Driver Code Ends
```

