

[Description](#)
[Solution](#)
[Discuss \(516\)](#)
[Submissions](#)

333. Largest BST Subtree

Medium
 1354
 107
 Add to List
 Share

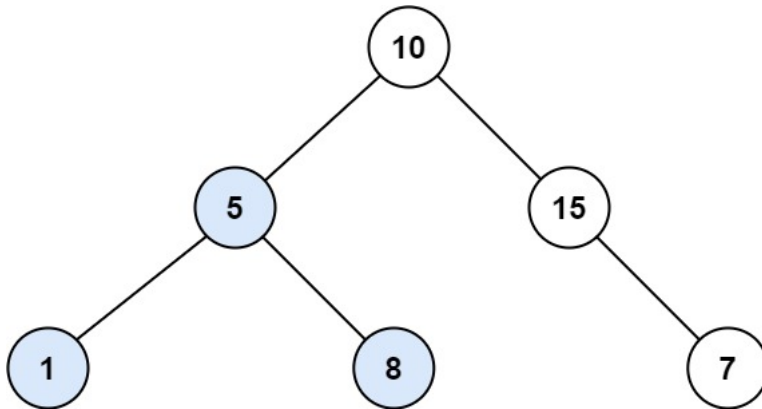
Given the root of a binary tree, find the largest subtree, which is also a Binary Search Tree (BST), where the largest means subtree has the largest number of nodes.

A **Binary Search Tree (BST)** is a tree in which all the nodes follow the below-mentioned properties:

- The left subtree values are less than the value of their parent (root) node's value.
- The right subtree values are greater than the value of their parent (root) node's value.

Note: A subtree must include all of its descendants.

Example 1:



Input: root = [10,5,15,1,8,null,7]

Output: 3

Explanation: The Largest BST Subtree in this case is the highlighted one. The return value is the subtree's size, which is 3.

Example 2:

Input: root = [4,2,7,2,3,5,null,2,null,null,null,null,1]

Output: 2

Constraints:

- The number of nodes in the tree is in the range $[0, 10^4]$.
- $-10^4 \leq \text{Node.val} \leq 10^4$

Follow up: Can you figure out ways to solve it with $O(n)$ time complexity?

Accepted 93,364 Submissions 219,193

Seen this question in a real interview before?

Companies [i](#)

Related Topics

Show Hint 1

```

1 //
2 * Definition for a binary tree
  node.
3 struct TreeNode {
4     int val;
5     TreeNode *left;
6     TreeNode *right;
7     TreeNode() : val(0),
  left(nullptr), right(nullptr) {}
8     TreeNode(int x) : val(x),
  left(nullptr), right(nullptr) {}
9     TreeNode(int x, TreeNode
    *left, TreeNode *right) : val(x),
  left(left), right(right) {}
10    * };
11    */
12    class Curr
13    {
14    public:
15        bool isBst;
16        int mini;
17        int maxi;
18        int nodes;
19
20        Curr(int one,int two,int
  three,bool ans)
21        {
22            isBst = ans;
23            mini = one;
24            maxi = two;
25            nodes = three;
26        }
27
28
29    };
30    class Solution {
31    public:
32        Curr largestTree(TreeNode*
  root,int &maxNodes)
33        {
34            if(!root)
35            {
36                return
  Curr(INT_MAX,INT_MIN,0,true);
37            }
38
39            if(!root->left and !root-
  >right)
40            {
41                Curr curr(root-
  >val,root->val,1,true);
42                maxNodes =
  max(maxNodes,1);
43                return curr;
44            }
45
  
```

NEW

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Your input [4,2,7,2,3,5,null,2,null,null,

Output 2 Diff

Expected 2

[Use Example](#)
[Testcases](#)

Run Code ^

Submit

Problems

Pick One

< Prev

333/2509

Next >

