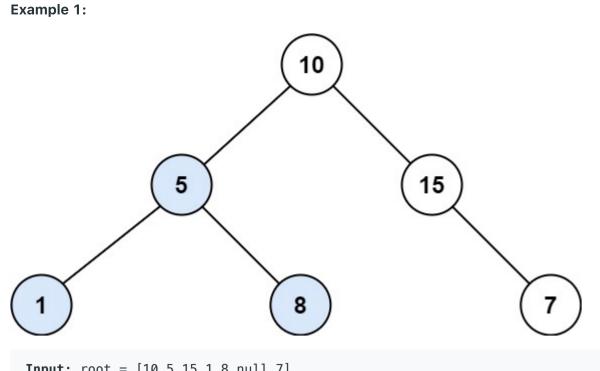
333. Largest BST Subtree

Medium ௴ 1000 ♀ 88 ♡ 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.



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:

Output: 2

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

Constraints:

• The number of nodes in the tree is in the range $[0, 10^4]$. • $-10^4 \le Node.val \le 10^4$

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

Accepted 72,270 Submissions 181,012 Seen this question in a real interview before? Yes No

Companies 🛅 i 0 ~ 6 months 6 months ~ 1 year ~ 1 year ~ 2 years

Amazon | 2 | ByteDance | 2

Dynamic Programming Tree Depth-First Search Binary Search Tree Binary Tree

Related Topics

Hide Hint 1 You can recursively use algorithm similar to 98. Validate Binary Search Tree at each node of the tree, which will result in O(nlogn) time complexity.

i Java

■ Autocomplete 1 ▼ /** * Definition for a binary tree node.

* public class TreeNode {

* int val;

* TreeNode left;

* TreeNode right; 7 * TreeNode() {} 8 * TreeNode(int val) { this.val = val; }
9 * TreeNode(int val, TreeNode left, TreeNode right) { 10 * this.val = val;

 X Pick One

 ✓ Prev
 ₹/99
 Next >
 Console →
 Contribute i
 ≡ Problems