

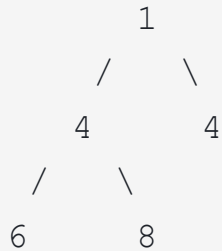
Size of Largest BST in Binary Tree

Given a binary tree. Find the size of its largest subtree that is a Binary Search Tree.

Note: Here Size is equal to the number of nodes in the subtree.

Example 1:

Input:

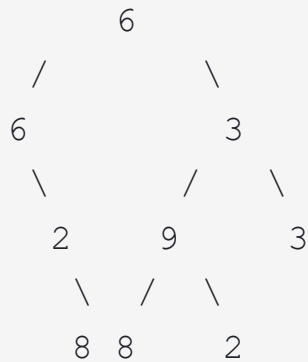


Output: 1

Explanation: There's no sub-tree with size greater than 1 which forms a BST. All the leaf Nodes are the BSTs with size equal to 1.

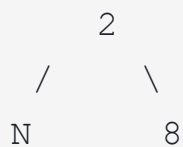
Example 2:

Input: 6 6 3 N 2 9 3 N 8 8 2



Output: 2

Explanation: The following sub-tree is a BST of size 2:



Your Task:

You don't need to read input or print anything. Your task is to complete the function **largestBst()** that takes the root node of the Binary Tree as its input and returns the size of the largest subtree which is also the BST. If the complete Binary Tree is a BST, return the size of the complete Binary Tree.

Expected Time Complexity: $O(N)$.

Expected Auxiliary Space: $O(\text{Height of the BST})$.

Constraints:

$1 \leq \text{Number of nodes} \leq 10^5$

$1 \leq \text{Data of a node} \leq 10^6$