Problem 1038: Binary Search Tree to Greater Sum Tree

Problem Information

Difficulty: Medium
Acceptance Rate: 88.33%
Paid Only: No
Tags: Tree, Depth-First Search, Binary Search Tree, Binary Tree

Problem Description

Given the `root` of a Binary Search Tree (BST), convert it to a Greater Tree such that every key of the original BST is changed to the original key plus the sum of all keys greater than the original key in BST.

As a reminder, a _binary search tree_ is a tree that satisfies these constraints:

* The left subtree of a node contains only nodes with keys **less than** the node's key. * The right subtree of a node contains only nodes with keys **greater than** the node's key. * Both the left and right subtrees must also be binary search trees.

Example 1:

Input: root = [4,1,6,0,2,5,7,null,null,null,3,null,null,null,8] **Output:** [30,36,21,36,35,26,15,null,null,null,33,null,null,null,8]

Example 2:

Input: root = [0,null,1] **Output:** [1,null,1]

Constraints:

* The number of nodes in the tree is in the range `[1, 100]`. * `0 <= Node.val <= 100` * All the values in the tree are **unique**.

Note: This question is the same as 538: <https://leetcode.com/problems/convert-bst-to-greater-tree/>

Code Snippets

C++:

```
/**
* Definition for a binary tree node.
* struct TreeNode {
* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode() : val(0), left(nullptr), right(nullptr) {}
* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
* TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
* };
*/
class Solution {
public:
TreeNode* bstToGst(TreeNode* root) {

}
};
```

Java:

```
/**
* Definition for a binary tree node.
* public class TreeNode {
* int val;
* TreeNode left;
* TreeNode right;
* TreeNode() {}
* TreeNode(int val) { this.val = val; }
* TreeNode(int val, TreeNode left, TreeNode right) {
* this.val = val;
```

```
* this.left = left;
* this.right = right;
* }
* }
*/
class Solution {
public TreeNode bstToGst(TreeNode root) {

}
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def bstToGst(self, root: Optional[TreeNode]) -> Optional[TreeNode]:
```