

538. Convert BST to Greater Tree

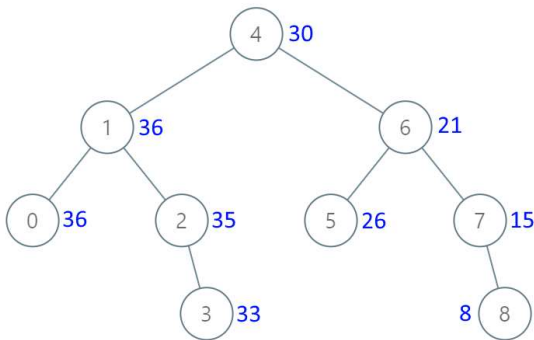
Medium 4154 158 Add to List Share

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 $[0, 10^4]$.
- $-10^4 \leq \text{Node.val} \leq 10^4$
- All the values in the tree are **unique**.
- `root` is guaranteed to be a valid binary search tree.

Note: This question is the same as 1038: <https://leetcode.com/problems/binary-search-tree-to-greater-sum-tree/>

Accepted 233,847 Submissions 354,877

Seen this question in a real interview before?

Companies

Related Topics

```
1  /**
2   * Definition for a binary tree node.
3   * public class TreeNode {
4   *     int val;
5   *     TreeNode left;
6   *     TreeNode right;
7   *     TreeNode() {}
8   *     TreeNode(int val) { this.val = val; }
9   *     TreeNode(int val, TreeNode left, TreeNode right) {
10  *         this.val = val;
11  *         this.left = left;
12  *         this.right = right;
13  *     }
14  * }
15  */
16  class Solution {
17  public TreeNode convertBST(TreeNode root) {
18
19  }
20 }
```