538. Convert BST to Greater Tree

Add to List

Description Submissions Solutions

- Total Accepted: **4583**
- Total Submissions: **8656**
- Difficulty: **Medium**
- Contributors:love_Fawn

Given 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 sum of all keys greater than the original key in BST.

Example:

Input: The root of a Binary Search Tree like this:

```
              5
            /   \
           2     13
```

Output: The root of a Greater Tree like this:

```
             18
            /   \
          20     13
```

Subscribe to see which companies asked this question.

Show Tags

Have you met this question in a real interview?

Yes

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    void inorderTree(TreeNode* root,vector<int> &res)
    {
        if(root==NULL) return;
        inorderTree(root->left,res);
        res.push_back(root->val);
        inorderTree(root->right,res);
    }

    void ChangeBST(TreeNode* root,vector<int> &treeval)
    {
        if(root==NULL) return; int index = -1;
        ChangeBST(root->left,treeval);
        for(int i=0;i<treeval.size();i++)
        {
            if(root->val == treeval[i])
            {
                index = i;
                break;
            }
        }
        int ans = accumulate(treeval.begin()+index,treeval.end(),0);
        root->val = ans;
        ChangeBST(root->right,treeval);
    }

    TreeNode* convertBST(TreeNode* root) {
        vector<int> treeval;
        inorderTree(root,treeval);
        ChangeBST(root,treeval);
```

```
        return root;
    }
};
```