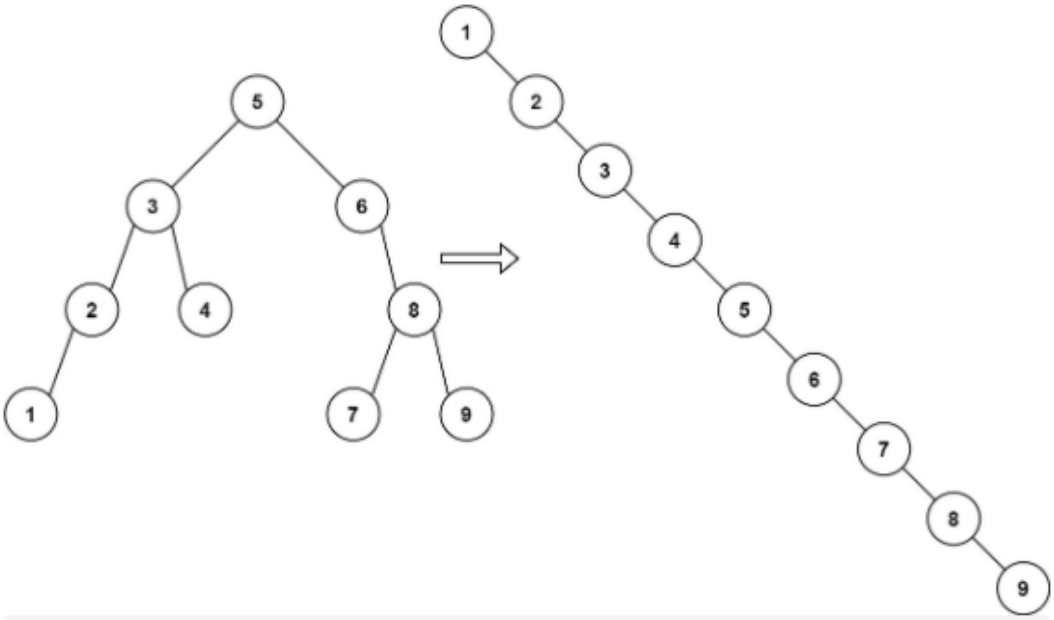
递增顺序搜索树

给你一棵二叉搜索树，请你 **按中序遍历** 将其重新排列为一棵递增顺序搜索树，使树中最左边的节点成为树的根节点，并且每个节点没有左子节点，只有一个右子节点。

示例 1：



输入：root = [5,3,6,2,4,null,8,1,null,null,null,7,9]

输出：[1,null,2,null,3,null,4,null,5,null,6,null,7,null,8,null,9]

```
/**
 * 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:
    void travel(TreeNode* root,vector<int>& v)
    {
        //中序遍历  左  根  右
        if(root==nullptr)
        {
```

```
            return;
        }
        travel(root->left,v);
        v.push_back(root->val);
        travel(root->right,v);
    }
    TreeNode* increasingBST(TreeNode* root) {
        vector<int> res;
        travel(root,res);
        TreeNode* newroot=new TreeNode(0);
        TreeNode* p=newroot;
        for(int i=0;i<res.size();i++)
        {
            p->right=new TreeNode(res[i]);
            p=p->right;
        }
        return newroot->right;
    }
};
```