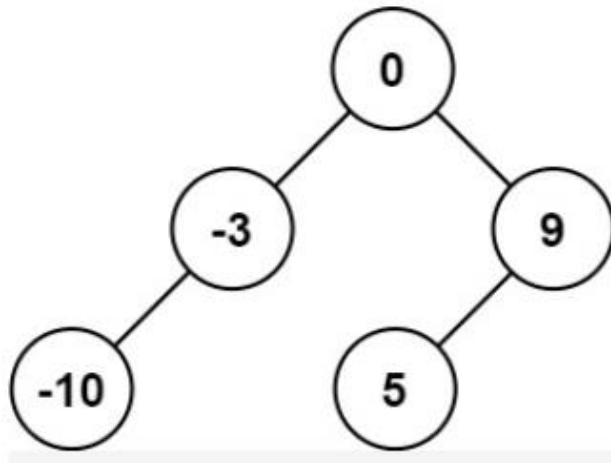


将有序数组转换为二叉搜索树

给你一个整数数组 `nums`，其中元素已经按 **升序** 排列，请你将其转换为一棵 **高度平衡** 二叉搜索树。

高度平衡 二叉树是一棵满足「每个节点的左右两个子树的高度差的绝对值不超过 1」的二叉树。

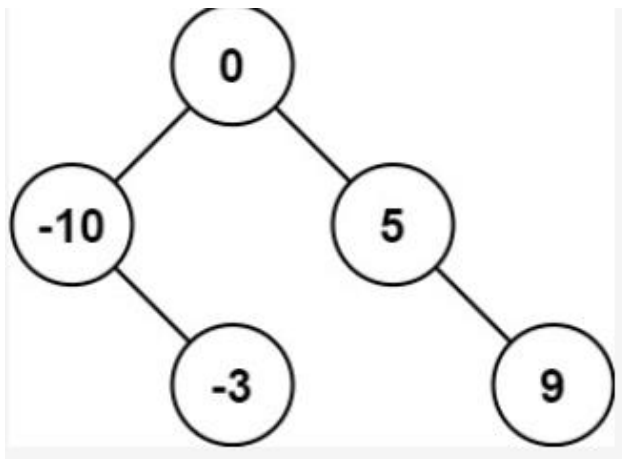
示例 1：



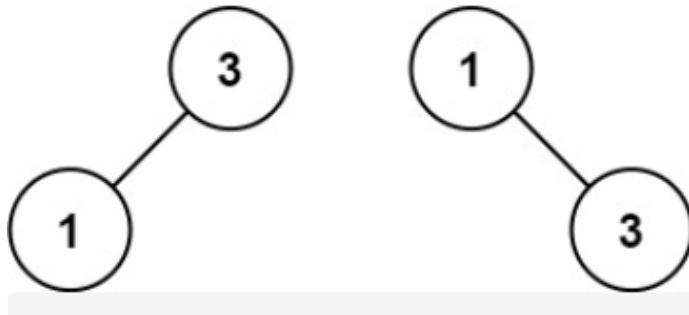
输入：nums = [-10,-3,0,5,9]

输出：[0,-3,9,-10,null,5]

解释：[0,-10,5,null,-3,null,9] 也将被视为正确答案：



示例 2：



输入: nums = [1,3]

输出: [3,1]

解释: [1,3] 和 [3,1] 都是高度平衡二叉搜索树。

```
/**
 * 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* travel(vector<int>& nums,int left,int right)
    {
        if(left>right)
            return nullptr;
        int mid=left+(right-left)/2;
        TreeNode* root=new TreeNode(nums[mid]);
        root->left=travel(nums,left,mid-1);
        root->right=travel(nums,mid+1,right);
        return root;
    }
    TreeNode* sortedArrayToBST(vector<int>& nums) {
        //二叉搜索树的中序遍历即为一个升序
        return travel(nums,0,nums.size()-1);
    }
};
```