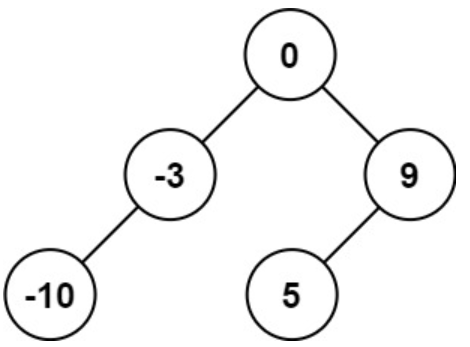
# 题目

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平衡二叉搜索树。

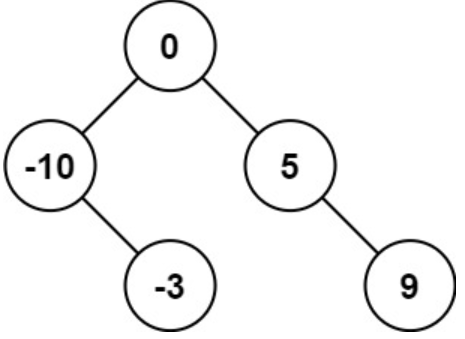
示例 1：



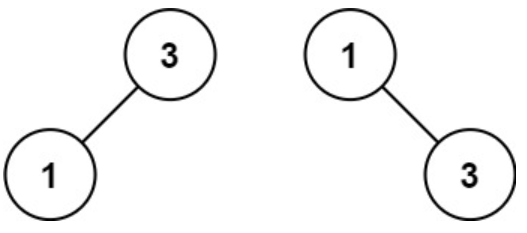
输入：nums = [-10,-3,0,5,9]

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,null,3]和[3,1]都是高度平衡二叉搜索树。

提示：

1 <= nums.length <= 10^4

-10^4 <= nums[i] <= 10^4

nums按严格递增顺序排列

# 分析

## 方法一：中序遍历

思路：

要将一个已按升序排列的整数数组转换为一棵平衡二叉搜索树，可以使用递归的方法。具体步骤如下：

1、找到数组的中间元素，将其作为根节点。

2、将数组分成两部分，左边部分用于构建左子树，右边部分用于构建右子树。

3、递归地构建左右子树。

代码：

class Solution {

public:

TreeNode\* sortedArrayToBST(vector<int>& nums) {

return helper(nums, 0, nums.size() - 1);

}

TreeNode\* helper(vector<int>& nums, int left, int right) {

if (left > right) {

return nullptr;

}

// 总是选择中间位置左边的数字作为根节点

int mid = (left + right) / 2;

TreeNode\* root = new TreeNode(nums[mid]);

root->left = helper(nums, left, mid - 1);

root->right = helper(nums, mid + 1, right);

return root;

}

};