# 题目

给定一个有序整数数组，元素各不相同且按升序排列，编写一个算法，创建一棵高度最小的二叉搜索树。

示例:

给定有序数组: [-10,-3,0,5,9],

一个可能的答案是：[0,-3,9,-10,null,5]，它可以表示下面这个高度平衡二叉搜索树：

0

/ \

-3 9

/ /

-10 5

# 分析

## 方法一：递归法/深度优先遍历

/\*\*

\* 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:

TreeNode\* dfs(const vector<int> &nums, int L, int R) {

if(L > R) {

return nullptr;

}

int mid = (L+R)>>1;

auto ptr = new TreeNode(nums[mid]); //填充根节点

ptr->left = dfs(nums, L, mid-1); //构造左子树

ptr->right = dfs(nums, mid+1, R); //构造右子树

return ptr;

}

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

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

}

};