108. Convert Sorted Array to Binary Search Tree

Easy

96499FavoriteShare

Given an array where elements are sorted in ascending order, convert it to a height balanced BST.

For this problem, a height-balanced binary tree is defined as a binary tree in which the depth of the two subtrees of *every* node never differ by more than 1.

Example:

Given the sorted array: [-10,-3,0,5,9],  
  
One possible answer is: [0,-3,9,-10,null,5], which represents the following height balanced BST:  
  
 0  
 / \  
 -3 9  
 / /  
 -10 5

Accepted

237,338

Submissions

482,664

/\*\*

\* 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 createBST(TreeNode\*& root, vector<int>& nums, int l, int h)

{

if (l <= h)

{

int m = l + (h - l) / 2;

root = new TreeNode(nums[m]);

createBST(root->left, nums, l, m - 1);

createBST(root->right, nums, m + 1, h);

}

}

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

TreeNode\* root = NULL;

createBST(root, nums, 0, nums.size() -1);

return root;

}

};