108. Convert Sorted Array to Binary Search Tree ★

Question | Editorial Solution

My Submissions (/problems/convert-sorted-array-to-binary-search-tree/submissions/)

Total Accepted: **92722** Total Submissions: **234381** Difficulty: **Medium** Contributors: **Admin**

Notes

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

Subscribe (/subscribe/) to see which companies asked this question

Show Tags

Show Similar Problems

Have you met this question in a real interview? Yes No

Discuss (https://leetcode.com/discuss/questions/oj/convert-sorted-array-to-binary-search-tree)

Pick One (/problems/random-one-question/)

C++

```
/**
 * 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 createTree(TreeNode* node, vector<int>::iterator lbegin, vector<int>::iter
        >::iterator rend) {
        auto lhalf = (lbegin + (lend - lbegin) / 2);
        auto rhalf = (rbegin + (rend - rbegin) / 2);
        node->left = new TreeNode(*lhalf);
        if(rhalf != rend)
            node->right = new TreeNode(*rhalf);
        if(lbegin != lhalf)
            createTree(node->left, lbegin, lhalf, lhalf + 1, lend);
        if(rbegin != rhalf)
            createTree(node->right, rbegin, rhalf, rhalf + 1, rend);
    }

    TreeNode* sortedArrayToBST(vector<int>& nums) {
        if (nums.empty()) return nullptr;
        TreeNode* root = new TreeNode(nums[nums.size() / 2]);
        if(nums.size() > 1)
            createTree(root, nums.begin(), nums.begin() + nums.size() / 2, nums.beg
        return root;
    }
```

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

Custom Testcase

Run Code | Submit Solution