链滴

[每日 LeetCode] 108. Convert Sorted Array to Binary Search Tree

作者：Hanseltu

原文链接：https://ld246.com/article/1559972405922

来源网站：链滴

原文链接 [每日LeetCode] 108. Convert Sorted Array to Binary Search Tree

Description:

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

For this problem, a height-balanced binary tree is defined as a binary tree in which the depth f 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
```

思路：本题要将已经排好序的数组构造成平衡二叉树。只需要保证每一次递归处理构造的左右子树节数量最多不超过1，就可以保证生成的二叉树满足高度平衡的性质。

因此，在递归时，每一次以中位数进行分割，将数组分割为左右两个部分并递归构造二叉树。

C++代码

```
/**
 * 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* sortedArrayToBST(vector<int>& nums) {
        return dfs(0, nums.size()-1, nums);
    }

    TreeNode* dfs(int left, int right, vector<int>& nums){
        int n = right - left + 1;
        if (n <= 0)
            return NULL;
        int m = n / 2;
        TreeNode* tRoot = new TreeNode(nums[left+m]);
```

```
        tRoot->left = dfs(left, left+m-1, nums);
        tRoot->right = dfs(left+m+1, right, nums);
        return tRoot;
    }
};
```

运行时间：16ms

运行内存：21.2M