链滴

[每日 LeetCode] 111. Minimum Depth of Binary Tree

作者：Hanseltu
原文链接：https://ld246.com/article/1559794579813
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

原文链接 [每日 LeetCode] 111. Minimum Depth of Binary Tree
Description:
Given a binary tree, find its minimum depth.
The minimum depth is the number of nodes along the shortest path from the root node own to the nearest leaf node.
Note: A leaf is a node with no children.
Example:
Given binary tree <code>[3,9,20,null,null,15,7]</code>,
<pre><code class="highlight-chroma">   3
   / \
  9 20
    / \
   15   7
</code></pre>
return its minimum depth = 2.
<hr>
思路：本题要求二叉树的最小深度，注意和 [每日 LeetCode] 104. Maximum Depth of Binary Tree 区别。本题依旧使用递归思想，需要注意的是在计算左子树和右子树深度的时候，判断是否等于 0，果等于 0，说明该子树不存在，深度赋值为最大值。
<hr>
C++ 代码
<pre><code class="highlight-chroma">/**
 * Definition for a inary tree node.
 * struct TreeNode {
 *   int val;
 *   TreeNode *lef;
 *   TreeNode *right;
 *   TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    int minDepth(TreeNode* root) {
        if(!root)
            return 0;
        if(!root->left && !root->right)
            return 1;
        int leftDepth = minDepth(root->left);
        int rightDept = minDepth(root->right);

        if(leftDepth
= 0)
            leftDepth
 INT_MAX;
        if(rightDepth
== 0)
            rightDepth
= INT_MAX;
        return leftDe
th < rightDepth ? (leftDepth + 1) : (rightDepth + 1);
    }
};
</code></pre>
<hr>
运行时间：8ms
运行内存：19.6M