# 题目

给定一个二叉树的 根节点root，请找出该二叉树的 最底层 最左边 节点的值。

假设二叉树中至少有一个节点。

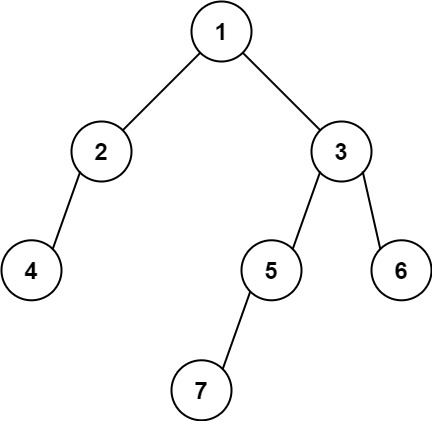
示例 1:



输入: root = [2,1,3]

输出: 1

示例 2:



输入: [1,2,3,4,null,5,6,null,null,7]

输出: 7

提示:

二叉树的节点个数的范围是 [1,104]

-231 <= Node.val <= 231 - 1

注意：本题与主站 513 题相同：

https://leetcode-cn.com/problems/find-bottom-left-tree-value/

# 分析

/\*\*

 \* Definition for a binary tree node.

 \* struct TreeNode {

 \*     int val;

 \*     TreeNode \*left;

 \*     TreeNode \*right;

 \*     TreeNode() : val(0), left(nullptr), right(nullptr) {}

 \*     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}

 \*     TreeNode(int x, TreeNode \*left, TreeNode \*right) : val(x), left(left), right(right) {}

 \* };

 \*/

class Solution {

public:

    int res = 0;

    int maxlevel = 0;

    int findBottomLeftValue(TreeNode\* root) {

        helper(root, 1);

        return res;

    }

    void helper(TreeNode\* root, int level){

        if(root == NULL) return;

        helper(root->left, level + 1);

        if(level > maxlevel){

            maxlevel = level;

            res = root->val;

        }

        helper(root->right, level + 1);

    }

};