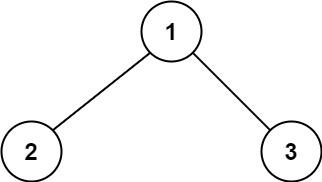
# 题目

路径被定义为一条从树中任意节点出发，沿父节点-子节点连接，达到任意节点的序列。同一个节点在一条路径序列中至多出现一次 。该路径至少包含一个节点，且不一定经过根节点。

路径和是路径中各节点值的总和。

给你一个二叉树的根节点 root ，返回其最大路径和。

示例 1：

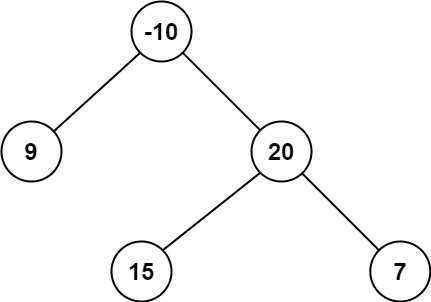


输入：root = [1,2,3]

输出：6

解释：最优路径是 2 -> 1 -> 3 ，路径和为 2 + 1 + 3 = 6

示例 2：



输入：root = [-10,9,20,null,null,15,7]

输出：42

解释：最优路径是 15 -> 20 -> 7 ，路径和为 15 + 20 + 7 = 42

提示：

树中节点数目范围是 [1, 3 \* 104]

-1000 <= Node.val <= 1000

# 分析

## 方法一：深度优先遍历

class Solution {

public:

int maxPathSum(TreeNode\* root) {

ret = INT\_MIN;

dfs(root);

return ret;

}

private:

int ret;

int dfs(TreeNode\* root) {

if (root == nullptr) {

return 0;

}

int left = max(0, dfs(root->left));

int right = max(0, dfs(root->right));

ret = max(ret, left + right + root->val);

return root->val + max(left, right);

}

};