# 题目

给定一棵二叉树，你需要计算它的直径长度。一棵二叉树的直径长度是任意两个结点路径长度中的最大值。这条路径可能穿过也可能不穿过根结点。

**示例：**

给定二叉树

1

/ \

2 3

/ \

4 5

返回 3，它的长度是路径[4,2,1,3]或者 [5,2,1,3]。

**注意：**两结点之间的路径长度是以它们之间边的数目表示。

# 分析

## 方法一：递归法

/\*\*

\* 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:

int ret;

int depth(TreeNode\* rt){

if (rt == NULL) return 0; // 访问到空节点了，返回0

int L = depth(rt->left); // 左儿子为根的子树的深度

int R = depth(rt->right); // 右儿子为根的子树的深度

ret = max(ret, L + R + 1); // 计算d\_node即L+R+1 并更新ans

return max(L, R) + 1; // 返回该节点为根的子树的深度

}

int diameterOfBinaryTree(TreeNode\* root) {

ret = 1;

depth(root);

return ret - 1;

}

};