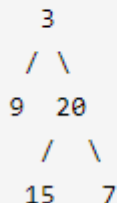


第4题——二叉树的深度

输入一棵二叉树的根节点，求该树的深度。从根节点到叶节点依次经过的节点（含根、叶节点）形成树的一条路径，最长路径的长度为树的深度。

例如：

给定二叉树 [3,9,20,null,null,15,7] ，



大佬的超级简洁解法（1）

```
int maxDepth(struct TreeNode* root){
    if (root == NULL) {
        return 0;
    }
    /* 当前节点的左子树的深度 */
    int lenLeft = maxDepth(root->left);
    /* 当前节点的右子树的深度 */
    int lenRight = maxDepth(root->right);
    /* 二叉树的深度等于左右子树深度的较大者加 1（1是当前节点的深度） */
    return lenLeft > lenRight ? lenLeft + 1 : lenRight + 1;
}
```

首先，如果树为空，此时树的深度为0，我们返回0；

其次，树没有孩子节点，那么会直接执行有三目运算符的return；此时，很明显，当前结点的左右子树的深度都为0；因此 lenLeft>lenRight不成立，取lenRight+1返回。

第三种，有孩子，则执行递归语句。