一、题目说明

题目543. Diameter of Binary Tree，计算二叉树的直径。直径是任意两个节点间的路径的最大值。难度是Easy!

二、我的解答

这个题目看懂不难，计算左子树的高度，右子树的高度，直径为二者之和。这里要注意的是，要计算每个节点的直径人，然后求最大直径，而不是单求树根的直径。

```
class Solution{
    public:
        int diameterOfBinaryTree(TreeNode* root){
            depth = 1;
            dfs(root);
            return depth-1;
        }
        //计算二叉树的深度
        int dfs(TreeNode* root){
            if(root == NULL) return 0;
            int left = dfs(root->left);
            int right = dfs(root->right);
            depth = max(left+right+1,depth);
            return max(left,right)+1;
        }
    private:
        int depth;
};
```

```
Runtime: 0 ms, faster than 100.00% of C++ online submissions for Diameter of Binary Tree.
Memory Usage: 19.7 MB, less than 92.59% of C++ online submissions for Diameter of Binary Tree.
```

三、优化措施

加上注释的代码：

```
class Solution{
    public:
        int diameterOfBinaryTree(TreeNode* root){
            dfs(root);
            return depth;
        }
        //计算二叉树的深度，类似后续遍历，表示以当前走到root为根节点，左右两边较长的路径
        int dfs(TreeNode* root){
            if(root == NULL) return 0;
            int left = dfs(root->left);
            int right = dfs(root->right);
            depth = max(left+right,depth);//计算节点直径，并更新depth的值，这里多加了个1
            return max(left,right)+1;
        }
```

```
    private:
        int depth=0;
};
```