递归版本/DFS

```
/*
struct TreeNode {
    int val;
    struct TreeNode *left;
    struct TreeNode *right;
    TreeNode(int x) :
            val(x), left(NULL), right(NULL) {
    }
};*/
class Solution {
public:
    int TreeDepth(TreeNode* pRoot)
    {
        if(pRoot==NULL)
            return 0;
        int left=TreeDepth(pRoot->left);
        int right=TreeDepth(pRoot->right);
        return (left>right?left:right)+1;
    }
};
```

非递归版本/BFS:

```
class Solution {
public:
    int TreeDepth(TreeNode* pRoot)
    {
        queue<TreeNode *>q;
        if(pRoot==NULL)
            return 0;
        q.push(pRoot);
        int depth=0;
        while(!q.empty())
        {
            int len=q.size();
            depth++;
            while(len--)
            {
                TreeNode * temp=q.front();
                q.pop();
                if(temp->left) q.push(temp->left);
                if(temp->right) q.push(temp->right);
            }
```

```
        }
        return depth;
    }
};
```