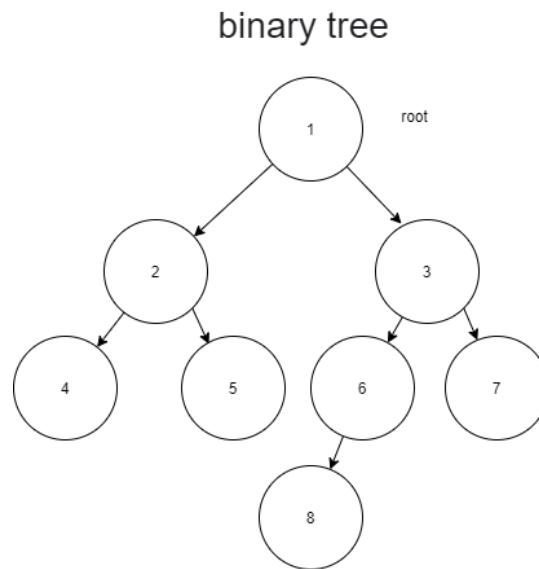


1. 用按层次顺序遍历二叉树的方法，统计树中具有度为1的结点数目。

解：下图为其树的形状：



下图为核心代码（可运行文件在 code 压缩包内，文件名为 HW8.1.cpp）

```
1 // 统计具有度为1的结点数目的函数
2 int count(TreeNode *root)
3 {
4     if (!root)
5     {
6         return 0;
7     }
8     // 创建一个队列用于存储待访问的节点
9     TreeNode **queue = (TreeNode **)malloc(sizeof(TreeNode *) * 100);
10    int front = 0, rear = 0; // 队列的前后指针
11    int count = 0;          // 度为1的结点数目
12    queue[rear++] = root;   // 将根节点入队
13    while (front < rear)
14    {
15        TreeNode *node = queue[front++]; // 取出队头节点
16        int degree = 0;
17        // 检查当前节点的左右子节点，如果存在则加入队列
18        if (node->left)
19        {
20            degree++;
21            queue[rear++] = node->left;
22        }
23        if (node->right)
24        {
25            degree++;
26            queue[rear++] = node->right;
27        }
28        // 如果当前节点的度为1，则将计数器加1
29        if (degree == 1)
30        {
31            count++;
32        }
33    }
34    free(queue); // 释放队列内存
35    return count;
36 }
37
38 }
```

2. 求任意二叉树中第一条最长的路径长度，并输出此路径上各结点的值。

解：二叉树的图形与 1. 一样。

图略

下图为核心代码（可运行文件在 code 压缩包内，文件名为 HW8.2.cpp）

```
1 void FindLongestPath(TreeNode *node, int length, int *MaxLength, int *currentPath, int *longestPath)
2 {
3     // 该节点不存在
4     if (node == NULL)
5         return;
6     // 该节点存在
7     else
8     {
9         // 先记录当前结点的值，随后长度加一
10        currentPath[length] = node->val;
11        length = length + 1;
12        // 已经到达叶子节点
13        if (node->left == NULL && node->right == NULL)
14        {
15            if (*MaxLength < length)
16            {
17                *MaxLength = length;
18                for (int i = 0; i < *MaxLength; i++) // 更新最长路径
19                {
20                    longestPath[i] = currentPath[i];
21                }
22            }
23            return;
24        }
25        // 还有子结点
26        else
27        {
28            FindLongestPath(node->left, length, MaxLength, currentPath, longestPath);
29            FindLongestPath(node->right, length, MaxLength, currentPath, longestPath);
30        }
31    }
32 }
```

3. 已知一棵度为 k 的树中有 n_1 个度为 1 的结点, n_2 个度为 2 的结点, ..., n_k 个度为 k 的结点, 问该树中有多少个叶子结点?

解: 二叉树的图形与 1. 一样。

图略

下图为核心代码 (可运行文件在 code 压缩包内, 文件名为 HW8. 3. cpp)

```
1 void LeafToRoot(TreeNode *node, int length, int *Path)
2 {
3     // 结点不存在
4     if (node == NULL)
5         return;
6     // 结点存在
7     else
8     {
9         Path[length] = node->val;
10        length = length + 1;
11        // 该结点为叶子结点
12        if (node->left == NULL && node->right == NULL)
13        {
14            // 输出路径上每个结点的值
15            for (int i = 0; i < length; i++)
16            {
17                printf("%d ", Path[i]);
18            }
19            printf("\n");
20        }
21        else
22        {
23            LeafToRoot(node->left, length, Path);
24            LeafToRoot(node->right, length, Path);
25        }
26    }
27 }
```

