

数据结构第五次实验报告

周依果、苏文泰、梁文俊

实验要求

第一部分

建立一个简单的人类图谱管理系统，实现如下功能：

- 分别对血缘遗传关系树进行存储和前序、中序及后序遍历；
- 分别以血缘遗传关系树为数据单元，把人类图谱森林存储在线性表中；
- 实现孩子及双亲的添加、查找和删除功能。

第二部分

用赫夫曼编码方法保存你班同学《数据结构》课程的期末考试成绩。

分析

第一部分

将实验要求抽象，我们需要建立一棵树这个数据结构来存储，其中树中各个节点的内容和之间的关系是由线性表来存储的。然后需要实现以下的功能

1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度（即记录的族谱的代数）
5. 修改节点值（修改记录的值）
6. 输出根节点（输出"记录的第一人"）
7. 输出某一节点的父节点（在二叉树族谱中为儿子）
8. 输出左右儿子（在二叉树族谱中为父母）
9. 增加子树
10. 删除子树

将实验要求抽象出来，我们便比较好明白我们需要做什么，需要写什么代码，而这些操作基本是树这种数据结构的基本操作，实现起来虽然代码量可能会比较大，不过合作起来就还好啦。

另外，由于为了方便调试和输入可能遇到的较为大批量的数据，我们采用文件读取的形式来获取数据，其中`TestData_Pre.txt`为初始化树的文件，`TestData_c1.txt`为子树的文件。

第二部分

实验要求过于直接，就是实现哈夫曼编码算法即可。

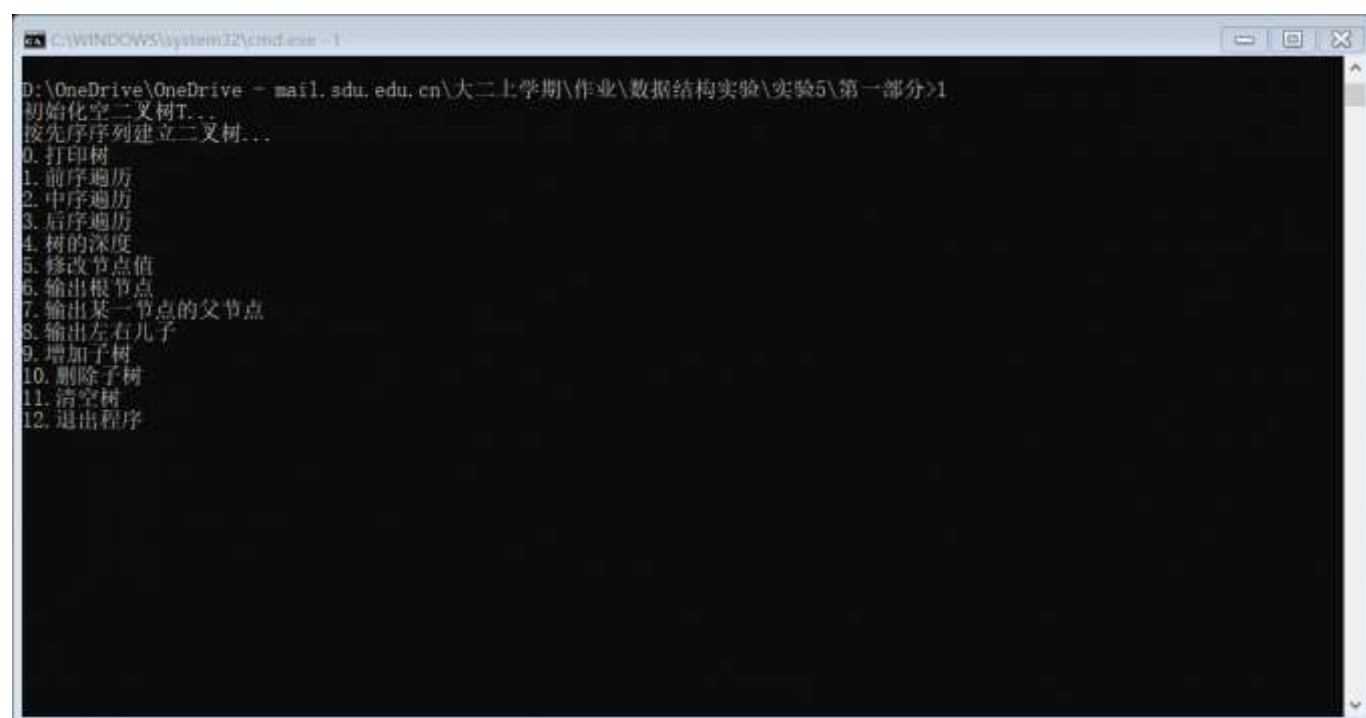
获取Huffman编码的基本思想是从Huffman树的叶子节点出发，自底向上若当前结点为父节点的左子树则在huffcode头部插入编码1，为左子树则在huffcode头部插入编码0（01顺序不影响最优编码长度），直到根节点，Huffman编码获取完成，从huffcode中顺序输出即可。

在实现Huffman树的难点在于如何从节点集合中找到两个权最小的节点并将其合并。然而在STL中的`priority_queue`基于小顶堆实现，能满足较快找到权重最小两节点的要求，我们于是采用优先队列来优化和简化代码实现。

代码实现

第一部分-二叉树

初始界面



```
C:\WINDOWS\system32\cmd.exe - 1

D:\OneDrive\OneDrive - mail.sdu.edu.cn\大二上学期\作业\数据结构实验\实验5\第一部分>1
初始化空二叉树T...
按先序序列建立二叉树...
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
```

0. 输出树的结构

```
C:\WINDOWS\system32\cmd.exe - 1
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
0
树的结构如下:
      A
     / \
    B   C
   / \   \
  D  E   F
 / \   \
G  H I   J

0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

1. 前序遍历

```
C:\WINDOWS\system32\cmd.exe - 1
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
1
前序遍历结果:
ABDGEHICFJ

0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

2. 中序遍历

```
C:\WINDOWS\system32\cmd.exe - 1
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
2
中序遍历结果:
GDBHEIAFJC
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

3. 后序遍历

```
C:\WINDOWS\system32\cmd.exe - 1
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
3
后序遍历结果:
GDHIEBJFCA
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

4. 树的深度（即记录的族谱的代数）

```
C:\WINDOWS\system32\cmd.exe - 1
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
4
二叉树 T 的深度为: 4
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

5. 修改节点值（修改记录的值）

```
C:\WINDOWS\system32\cmd.exe - 1
请输入操作:
5
请输入要修改的节点
F
请输入修改后的名称
X
将结点 F 修改为 X 后, T =
      A
     / \
    B   C
   / \ / \
  D E X H I J
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

6. 输出根节点（输出"记录的第一人"）

```
C:\WINDOWS\system32\cmd.exe - 1
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
6
T的根结点为 A
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

7. 输出某一节点的父节点 （在二叉树族谱中为儿子）

```
C:\WINDOWS\system32\cmd.exe - 1
12. 退出程序
请输入操作:
7
请输入要查找的节点
X
结点 X 的父节点为: C
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

8. 输出左右儿子 （在二叉树族谱中为父母）

```
C:\WINDOWS\system32\cmd.exe - 1
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
8
请输入要查找的节点
X
结点 X 的左孩子结点值为: , 右孩子结点值为: J
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

9. 增加子树

```
C:\WINDOWS\system32\cmd.exe - 1
12. 退出程序
请输入操作:
9
请输入要插入的叶子点
X
创建子树 c1 ...
选择插入左子树or右子树:
0. 左子树
1. 右子树
1
将子树 c1 插入为二叉树 T 中 X 结点的右子树 ...
      B   C
     D E X
    G H I J
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
```

10. 删除子树

```
C:\WINDOWS\system32\cmd.exe - 1
12. 退出程序
请输入操作:
10
请输入要删除的叶子点
X
选择删除左子树or右子树:
0. 左子树
1. 右子树
1
删除二叉树 T 中 X 结点的右子树 ...
      A
     / \
    B   C
   / \
  D   E
 / \
G   H I

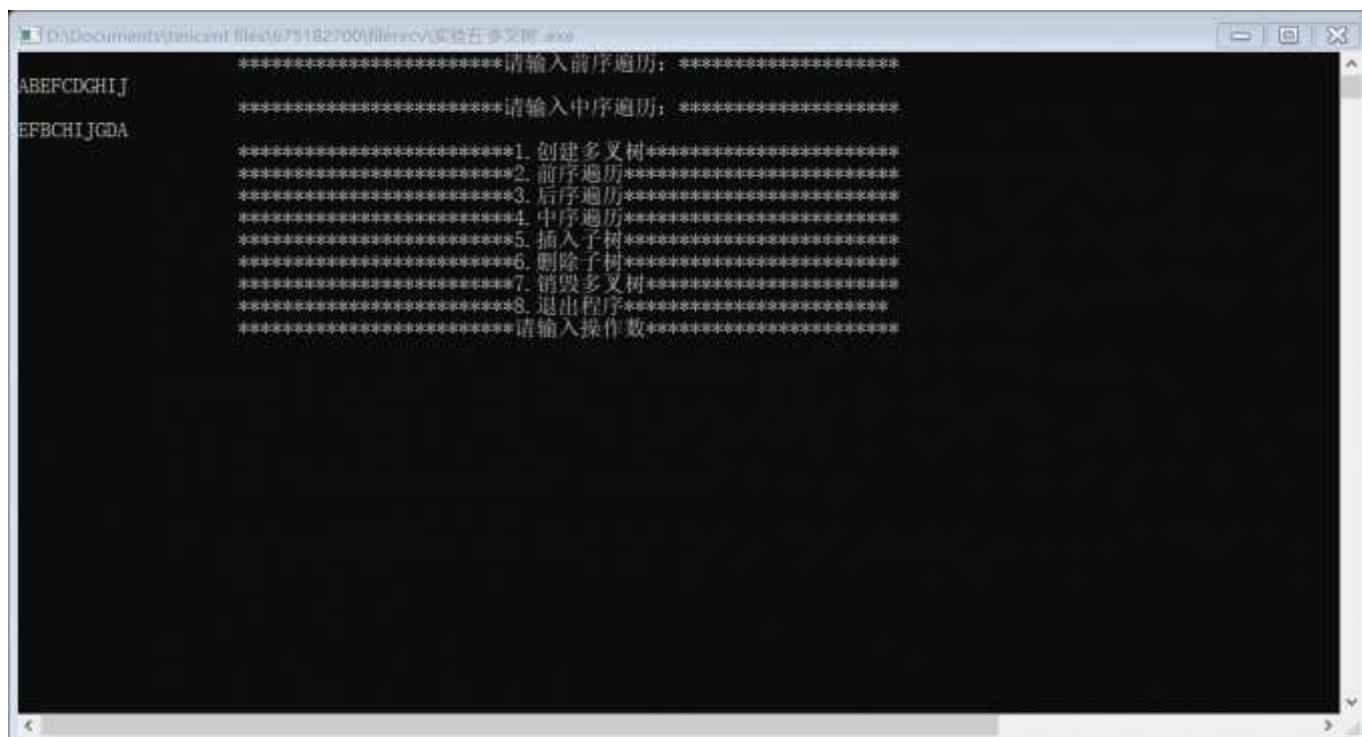
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

11. 删除整个树

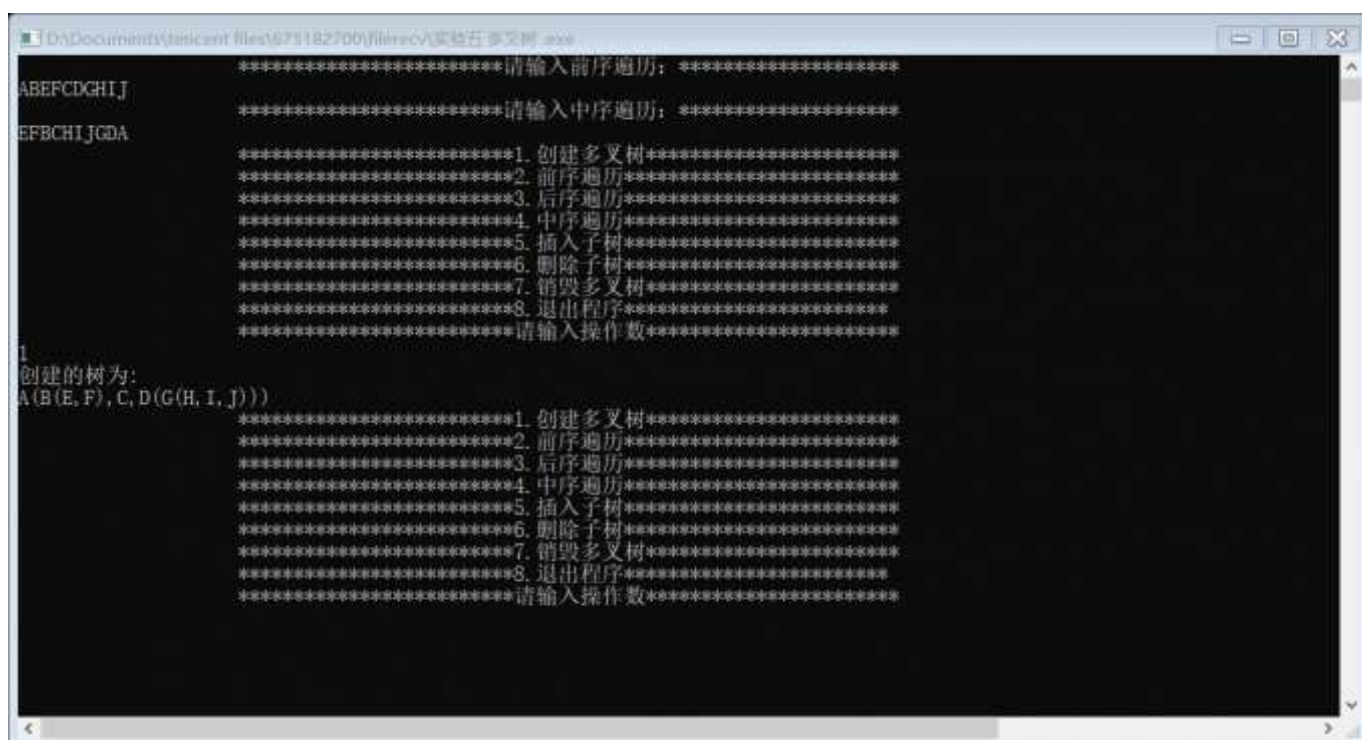
```
C:\WINDOWS\system32\cmd.exe - 1
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
11
T 为空!
0. 打印树
1. 前序遍历
2. 中序遍历
3. 后序遍历
4. 树的深度
5. 修改节点值
6. 输出根节点
7. 输出某一节点的父节点
8. 输出左右儿子
9. 增加子树
10. 删除子树
11. 清空树
12. 退出程序
请输入操作:
```

第一部分-多叉树

0. 初始化



1. 创建多叉树



2. 前序遍历

```
D:\Documents\tiancent files\675182700\filesrc\八宝姑丹 多叉树.exe

*****5. 插入子树*****
*****6. 删除子树*****
*****7. 销毁多叉树*****
*****8. 退出程序*****
*****请输入操作数*****

1
创建的树为:
A(B(E, F), C, D(G(H, I, J)))
*****1. 创建多叉树*****
*****2. 前序遍历*****
*****3. 后序遍历*****
*****4. 中序遍历*****
*****5. 插入子树*****
*****6. 删除子树*****
*****7. 销毁多叉树*****
*****8. 退出程序*****
*****请输入操作数*****

2
前序遍历:
A B E F C D G H I J
*****1. 创建多叉树*****
*****2. 前序遍历*****
*****3. 后序遍历*****
*****4. 中序遍历*****
*****5. 插入子树*****
*****6. 删除子树*****
*****7. 销毁多叉树*****
*****8. 退出程序*****
*****请输入操作数*****
```

3. 后序遍历

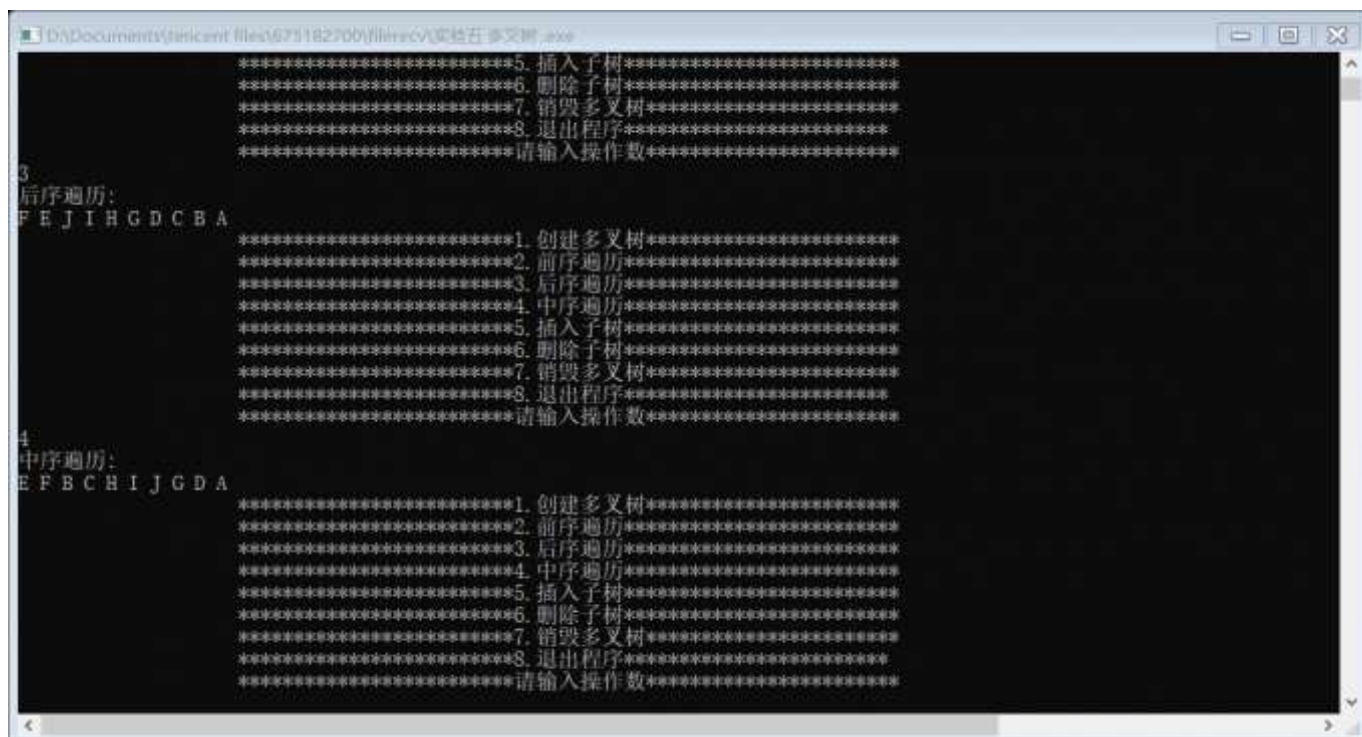
```
D:\Documents\tiancent files\675182700\filesrc\八宝姑丹 多叉树.exe

*****5. 插入子树*****
*****6. 删除子树*****
*****7. 销毁多叉树*****
*****8. 退出程序*****
*****请输入操作数*****

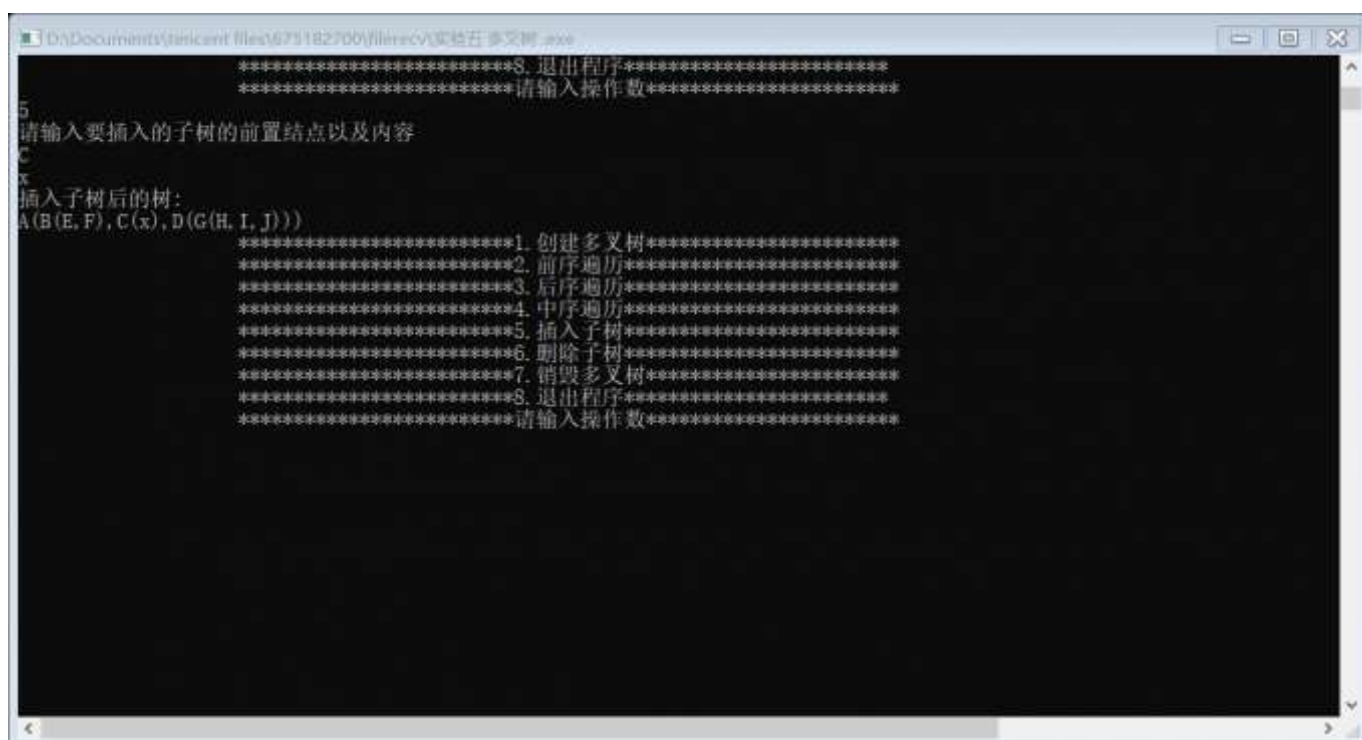
2
前序遍历:
A B E F C D G H I J
*****1. 创建多叉树*****
*****2. 前序遍历*****
*****3. 后序遍历*****
*****4. 中序遍历*****
*****5. 插入子树*****
*****6. 删除子树*****
*****7. 销毁多叉树*****
*****8. 退出程序*****
*****请输入操作数*****

3
后序遍历:
F E J I H G D C B A
*****1. 创建多叉树*****
*****2. 前序遍历*****
*****3. 后序遍历*****
*****4. 中序遍历*****
*****5. 插入子树*****
*****6. 删除子树*****
*****7. 销毁多叉树*****
*****8. 退出程序*****
*****请输入操作数*****
```

4. 中序遍历



5. 插入子树



6. 删除子树

```
D:\Documents\tencent files\675182700\files\c\实验五\多叉树.exe
插入子树后的树:
A(B(E,F),C(x),D(G(H,I,J)))
*****1. 创建多叉树*****
*****2. 前序遍历*****
*****3. 后序遍历*****
*****4. 中序遍历*****
*****5. 插入子树*****
*****6. 删除子树*****
*****7. 销毁多叉树*****
*****8. 退出程序*****
*****请输入操作数*****
6
请输入要删除的子树
x
删除后, 树为:
A(B(E,F),C,D(G(H,I,J)))
*****1. 创建多叉树*****
*****2. 前序遍历*****
*****3. 后序遍历*****
*****4. 中序遍历*****
*****5. 插入子树*****
*****6. 删除子树*****
*****7. 销毁多叉树*****
*****8. 退出程序*****
*****请输入操作数*****
```

第二部分

输入学生数量，再分别输入学生姓名和成绩，即可。

```
C:\WINDOWS\system32\cmd.exe
D:\OneDrive\OneDrive - mail.sdu.edu.cn\大二上学期\作业\数据结构实验\实验5\第二部分>1
Please enter the number of the student:
6
Please enter the name and grade of the students:
abc 45
dsk 99
skd 88
dsk 90
odi 77
ods 120
The grade is not in 0 ~ 100, please try again
ods 20
abc's grade is:45 and Huffman code is:1101
dsk's grade is:99 and Huffman code is:10
skd's grade is:88 and Huffman code is:00
dsk's grade is:90 and Huffman code is:01
odi's grade is:77 and Huffman code is:111
ods's grade is:20 and Huffman code is:1100
D:\OneDrive\OneDrive - mail.sdu.edu.cn\大二上学期\作业\数据结构实验\实验5\第二部分>
```

程序运行结果

运行结果正常，较好的完成了实验的需求和目标。