

实验 3.1 二叉树基本操作

此项实验主要测试对二叉树的一些基本操作，包括：初始化、销毁、构造、求其深度、插入结点、输出符合条件的结点、交换所有结点的左右孩子以及非递归遍历。

1. 初始化

```
E:\Microsoft Visual Studio\project\Projects\实验\Debug\实验3.1.exe
1-初始化二叉树
2-根据扩展先序遍历序列创建二叉链表
3-求二叉树的深度
4-插入结点
5-按中序遍历二叉树
6-输出叶结点
7-输出度为1的结点
8-输出度为2的结点
9-交换所有结点的左右孩子结点
10-按树状打印二叉树
11-销毁二叉树
101-退出
Enter choice:1
初始化完成!
请按任意键继续. . .
```

2. 构造二叉树

```
Enter choice:2
请输入二叉树扩展先序遍历序列(用'.'表示空子树)<如:AB..CD...>:
A
B
.
D
F
.
G
.
C
.
E
.
H
.
请按任意键继续. . .
```

根据扩展先序遍历序列创建二叉树，用“.”表示空子树。

3. 求二叉树深度

```
Enter choice:3
该二叉树的深度为: 4
请按任意键继续. . .
```

4. 插入结点

```
Enter choice:4
请输入目标结点元素:H
请输入插入结点元素:I
请输入需要将该结点插做左或右孩子结点(偶数为左孩子、奇数为右孩子):6
请按任意键继续. . .
Enter choice:10
      H
     / \
    I   .
   / \
  E   .
 / \
C   .
/ \
A   .
 / \
G   .
/ \
D   .
/ \
F   .
/ \
B   .
打印完成!
```

先输入目标结点元素，找到目标结点，然后再输入待插入的结点元素，再输入要插为左孩子还是右孩子（偶数为左孩子，奇数为右孩子）。

5. 中序遍历二叉树（非递归遍历）

```
Enter choice:5
中序遍历后的二叉树为:B F D G A C E I H
请按任意键继续. . .
```

6. 输出叶结点

```
Enter choice:6
叶结点为:F G I
请按任意键继续. . .
```

7. 输出度为 1 的结点

```
Enter choice:7
度为1的结点: B C E H
请按任意键继续. . .
```

8. 输出度为 2 的结点

```
Enter choice:8
度为2的结点为:A D
请按任意键继续. . .
```

9. 交换左右孩子结点

```
Enter choice:10
  H
  I
 E
 C
A
  G
  D
  F
  B
打印完成!
请按任意键继续. . .
Enter choice:9
交换完成!
请按任意键继续. . .
Enter choice:10
  B
  F
  D
  G
A
  C
  E
  I
  H
打印完成!
请按任意键继续. . .
```

10. 销毁二叉树

```
Enter choice:11
已销毁!
请按任意键继续. . .
```

实验 3.2 Huffman 编码

此实验主要实现建立 Huffman 树及 Huffman 编码的相关操作。包括：初始化 Huffman 树、建立 Huffman 树、Huffman 编码及 Huffman 译码，还有将相关数据写进文件以及将文件的内容打印到终端上。

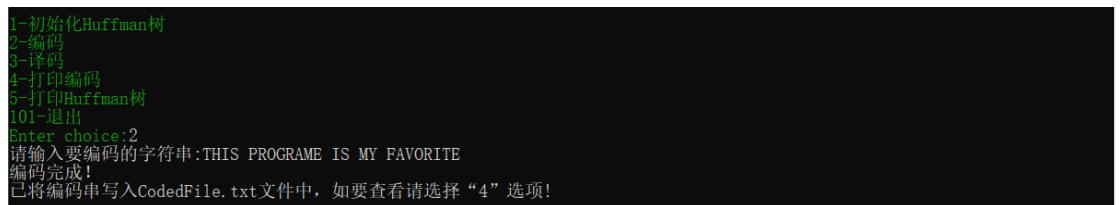
1. 初始化 Huffman 树



```
E:\Microsoft Visual Studio\project\Projects\实验\Debug\实验3.2.exe
*****主菜单*****
1-初始化Huffman树
2-编码
3-译码
4-打印编码
5-打印Huffman树
101-退出
Enter choice:1
请输入字符集数: 27
Huffman树已建成!
已将Huffman树写入HuffmanTree.txt文件中, 如要查看请选择“5”选项!
```

输入字符集数：27，包括空格和 26 个大写字母，对应的权值已写进对应的文件，建立 Huffman 树是自动调用，然后将建好的 Huffman 树写进 Huffman.txt 文件中，选“5”选项即可查看。

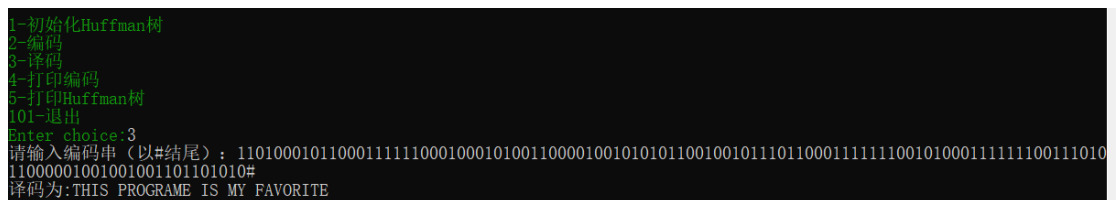
2. Huffman 编码



```
1-初始化Huffman树
2-编码
3-译码
4-打印编码
5-打印Huffman树
101-退出
Enter choice:2
请输入要编码的字符串:THIS PROGRAME IS MY FAVORITE
编码完成!
已将编码串写入CodedFile.txt文件中, 如要查看请选择“4”选项!
```

将编码串存放到 CodedFile.txt 文件中，选“4”选项可查看。

3. 译码



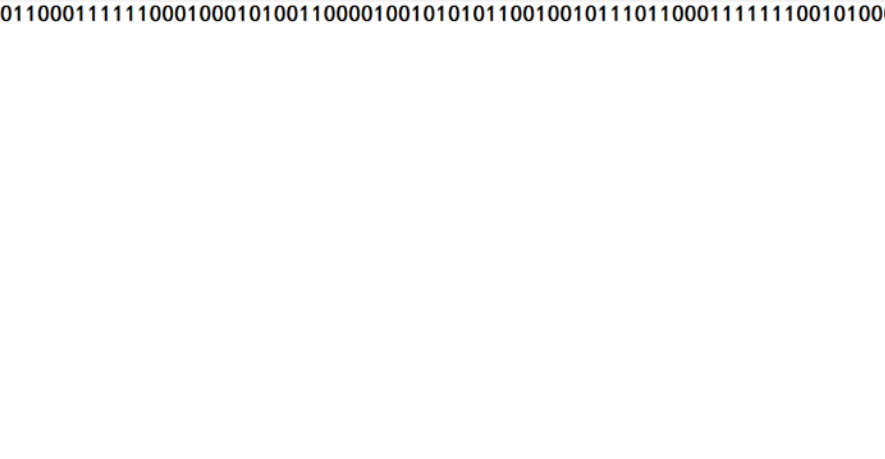
```
1-初始化Huffman树
2-编码
3-译码
4-打印编码
5-打印Huffman树
101-退出
Enter choice:3
请输入编码串(以#结尾): 110100010110001111111000100010100110000100101010110010010111011000111111100101000111111100111010
1100000100100100110110101010#
译码为:THIS PROGRAME IS MY FAVORITE
```

对 CodedFile.txt 文件中的编码串进行解码。

4. 将第二步输入到 CodedFile.txt 中的编码串打印到终端（每行打印 50 个代码）

```
1-初始化Huffman树
2-编码
3-译码
4-打印编码
5-打印Huffman树
101-退出
Enter choice:4
编码串为:
110100001011000111111000100010100110000100101010110
101001011101000111111001010001111110011101011000
0010010010011101101010
```

CodedFile.txt 文件:

A screenshot of a Notepad window titled "CodedFile.txt - 记事本". The window contains a single line of binary code: "1101000101110001111110001000101001100001001010101100100101110110001111111001010001111111". The status bar at the bottom indicates "第 1 行, 第 1 列" (Line 1, Column 1), "100%", "Windows (CRLF)", and "UTF-8".


CodedFile.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

1101000101110001111110001000101001100001001010101100100101110110001111111001010001111111

第 1 行, 第 1 列 100% Windows (CRLF) UTF-8

5. 将第一步输入到 Huffman.txt 中的 Huffman 书打印到终端



```
选择E:\Microsoft Visual Studio\project\Projects\实验\Debug\实验3.2.exe
Enter choice:5
Huffman树为:
结点i 字符 权值 双亲 左孩子 右孩子
1      186 50 0 0
2      A 64 45 0 0
3      B 13 33 0 0
4      C 22 37 0 0
5      D 32 39 0 0
6      E 103 48 0 0
7      F 21 36 0 0
8      G 15 33 0 0
9      H 47 41 0 0
10     I 57 43 0 0
11     J 1 28 0 0
12     K 5 31 0 0
13     L 32 39 0 0
14     M 20 36 0 0
15     N 57 43 0 0
16     O 63 44 0 0
17     P 15 34 0 0
18     Q 1 28 0 0
19     R 48 42 0 0
20     S 51 42 0 0
21     T 80 46 0 0
22     U 23 37 0 0
23     V 8 32 0 0
24     W 18 35 0 0
25     X 1 29 0 0
26     Y 16 34 0 0
27     Z 1 29 0 0
```

Huffman.txt 文件:

HuffmanTree.txt - 记事本					
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)					
结点i	字符	权值	双亲	左孩子	右孩子
1		186	50	0	0
2	A	64	45	0	0
3	B	13	33	0	0
4	C	22	37	0	0
5	D	32	39	0	0
6	E	103	48	0	0
7	F	21	36	0	0
8	G	15	33	0	0
9	H	47	41	0	0
10	I	57	43	0	0
11	J	1	28	0	0
12	K	5	31	0	0
13	L	32	39	0	0
14	M	20	36	0	0
15	N	57	43	0	0
16	O	63	44	0	0
17	P	15	34	0	0
18	Q	1	28	0	0
19	R	48	42	0	0
20	S	51	42	0	0

第 1 行, 第 1 列100%Windows (CRLF)ANSI