题目：建立霍夫曼树并实现编码和译码。

班级：电信1808班 姓名：郭佳璐 学号：U201813461 完成日期：2019年4月8日

**一．需求分析**

1. 初始化（ Createtree）。从终端读入27个字符的权值，建立哈夫曼树。
2. 编码（Hfmcode）。利用已经建好的哈夫曼树对27个字符进行编码并存入结构体codetype中。
3. 译码（Decode）。从终端读入代码并进行读码同时存入 字符串ToBeTran中。
4. 打印出代码的编码和译码。
5. 测试数据。

输入空格和A~Z字母的频度：

186 64 13 22 32 103 21 15 47 57 1 5 32 20 57 63 15 1 48 51 80 23 8 18 1 16 1

请输入需要编码的字符串：

THIS PROGRAM IS MY FAVORITE

该字符串编码为：

1101000101100011111100010001010011000010010101011001011101100011111110010100011111110011101011000001001001001101101010

输入需要译文的编码（以#号结束）：

1101000101100011111100010001010011000010010101011001011101100011111110010100011111110011101011000001001001001101101010#

编码译文为：

THIS PROGRAM IS MY FAVORITE

**二．概要设计。**

1. 本程序包括四个模块
2. （1）主程序模块：

Int main()

{

初始化；

接受命令；

处理命令；

}

1. 初始化模块：

构建树的结构体（DATA）和储存编码的结构体（codetype）。

1. 构建霍夫曼树模块（Createtree）：

构建哈夫曼树，传数组hfmTree和字符个数N做参数。

1. 编码模块（Hfmcode）：

对输入的字符串利用已经建好的霍夫曼树进行编码。

1. 译码模块（Decode）：

对输入的代码进行解码。

**三．详细设计。**

1. 初始化

typedef struct

{

char CH;//字符

int weight;//权值

int parent, lchild, rchild;//双亲，左孩子，右孩子

}DATA;//树的结构体

typedef struct

{

char code[30];

int cnt;

}codetype;

1. 建树模块

void Createtree(DATA \*hfmTree, int N)；

1. 编码模块

void Hfmcode(DATA \*hfmTree, codetype \*codeFile, int N)

{

int i, p, c;

codetype S;

for (i = 0; i<N; i++) {

c = i;

p = hfmTree[c].parent;

S.cnt = N;

S.code[N] = '\0';

while (p != -1)

{

if (hfmTree[p].lchild == c)；

S.code[--S.cnt] = '0';

else

S.code[--S.cnt] = '1';

c = p;

p = hfmTree[c].parent;

}

codeFile[i] = S;

}

1. 译码模块

void Decode(DATA \*hfmTree,char \*ToBeTran, int N)

{

int i,ct=0;

char ch;

scanf("%c", &ch);

i = 2 \* N - 2;

while (ch!='#')

{

if (ch == '0')

i = hfmTree[i].lchild;

else if (ch == '1')

i = hfmTree[i].rchild;

if (hfmTree[i].lchild == -1 || hfmTree[i].rchild == -1)

{

ToBeTran[ct++] = hfmTree[i].CH;

i = 2 \* N - 2;

}

scanf("%c", &ch);

}

if ((hfmTree[i].lchild != -1 || hfmTree[i].rchild != -1) && i != 2 \* N - 2)

printf("编码有误！");

ToBeTran[ct] = '\0';

}

1. 主函数模块

int main()

{

int N=27;

int i, j;

char str[200];

char \*ToBeTran,c;

DATA \*hfmTree;

codetype \*codeFile;

ToBeTran = (char \*)malloc(sizeof(char) \* 40);

codeFile = (codetype \*)malloc(sizeof(codetype)\*N);

hfmTree = (DATA \*)malloc(sizeof(DATA)\*(2 \* N - 1));

printf("输入空格和A~Z字母的频度：\n");

Createtree(hfmTree, N);

Hfmcode(hfmTree, codeFile, N);

scanf("%c", &c);

printf("请输入需要编码的字符串：\n");

gets(str);

printf("\n");

printf("该字符串编码为：\n");

for (i = 0; i < strlen(str); i++)

{

if (str[i] == ' ')

printf("%s", codeFile[0].code + codeFile[0].cnt);

else

printf("%s", codeFile[str[i] - 'A' + 1].code + codeFile[str[i] - 'A' + 1].cnt);

}

printf("\n\n");

printf("输入需要译文的编码（以#号结束）：\n");

Decode(hfmTree, ToBeTran, N);

printf("\n");

printf("编码译文为：\n");

printf("%s", ToBeTran);

return 0;

}

**四．调试分析**

该程序从键盘介入空格和26个字符的权值，并输入要编码的字符串，和要译码的代码，即可得到编码结果和译码结果。

**五．用户手册**

该程序从键盘介入空格和26个字符的权值，并输入要编码的字符串，和要译码的代码，即可得到编码结果和译码结果。

1. **调试结果**

输入空格和A~Z字母的频度：

186 64 13 22 32 103 21 15 47 57 1 5 32 20 57 63 15 1 48 51 80 23 8 18 1 16 1

请输入需要编码的字符串：

THIS PROGRAM IS MY FAVORITE

该字符串编码为：

1101000101100011111100010001010011000010010101011001011101100011111110010100011111110011101011000001001001001101101010

输入需要译文的编码（以#号结束）：

1101000101100011111100010001010011000010010101011001011101100011111110010100011111110011101011000001001001001101101010#

编码译文为：

THIS PROGRAM IS MY FAVORITE

**七．附录**

DATA 树的结构体

m = 0 c = 0 记录最小值和次小值的下标