**题目10：霍夫曼编码应用**

假设一串由大写字母构成的电文，采用霍夫曼规则对其进行编码，以菜单方式设计并完成功能任务：建立霍夫曼树、霍夫曼编码生成、编码文件译码。

**代码：**

#include<stdio.h>

#include<string.h>

#include<math.h>

#define n 100

#define m 2\*n-1

typedef struct //码结点的存储结构

{

char ch;

char bits[9];

int len;

}CodeNode;

typedef CodeNode HuffmanCode[n+1];

typedef struct //树节点的存储结构

{

int weight;

int lchild,rchild,parent;

}HTNode;

typedef HTNode HuffmanTree[m+1];

int num=0;

void select(HuffmanTree HT,int k,int &s1,int &s2) //挑选权值最小的两个结点

{

int i,j;

int min1=32767;

for(i=1;i<=k;i++)

if(HT[i].weight<min1 && HT[i].parent==0)

{

j=i;

min1=HT[i].weight;

}

s1=j;min1=32767;

for(i=1;i<=k;i++)

if(HT[i].weight<min1&&HT[i].parent==0&&i!=s1)

{

j=i;

min1=HT[i].weight;

}

s2=j;

}

int jsq(char \*s,int cnt[],char str[]) //统计输入字符和串

{

char \*p;

int i,j,k=0;

int temp[257];

for(i=0;i<257;i++)

temp[i]=0;

for(p=s;\*p!='\0';p++)

temp[\*p]++;

for(i=0,j=0;i<=256;i++)

if(temp[i]!=0)

{

j++;

str[j]=i;

cnt[j]=temp[i];

}

num=j;

return j;

}

//建立霍夫曼树

void chuffmanTree(HuffmanTree&HT,HuffmanCode&HC,int cnt[],char str[])

{

int i,s1,s2;

for(i=1;i<=2\*num-1;i++)

{

HT[i].lchild=0;

HT[i].rchild=0;

HT[i].parent=0;

HT[i].weight=0;

}

for(i=1;i<=num;i++)

HT[i].weight=cnt[i];

for(i=num+1;i<=2\*num-1;i++)

{

select(HT,i-1,s1,s2);

HT[s1].parent=i;

HT[s2].parent=i;

HT[i].lchild=s1;

HT[i].rchild=s2;

HT[i].weight=HT[s1].weight+HT[s2].weight;

}

for(i=1;i<=num;i++)

HC[i].ch=str[i];

}

//给霍夫曼树的每个叶子结点分配二进制代码

void HuffmanEncoding(HuffmanTree HT,HuffmanCode HC)

{

int c,p,i;

char cd[n];

int start;

cd[num]='\0';

for(i=1;i<=num;i++) //1到num为叶子结点，每个节点都有二进制编码串

{

start=num;

c=i;

while((p=HT[c].parent)>0)

{

start--;

cd[start]=(HT[p].lchild==c)?'0':'1';

c=p;

}

strcpy(HC[i].bits,&cd[start]);

HC[i].len=num-start;

}

}

void decode(HuffmanCode HC,char receive[],char s[]) //译码

{

char str[268];

char cd[9];

int i,j,k=0,p=0,cjs;

while(receive[p]!='\0')

{

cjs=0;

for(i=0;i<num&&cjs==0&&receive[p]!='\0';i++)

{

cd[i]='\0';

cd[i+1]='\0';

cd[i]=receive[p++];

for(j=1;j<=num;j++)

if(strcmp(HC[j].bits,cd)==0)

{

str[k]=HC[i].ch;

k++;

cjs=1;

break;

}

}

}

str[k]='\0';

strcpy(s,str);

}

void coding(HuffmanCode HC,char str[],char get[]) //输出字符串的二进制编码

{

int i,j=0;

while(str[j]!='\0')

{

for(i=1;i<=num;i++)

if(HC[i].ch==str[j])

{

strcat(get,HC[i].bits);

break;

}

j++;

}

strcat(get,"\0");

}

int main()

{

char str[257]; //str用于在统计输入序列中的字母时用，存放是什么字符，1到256

char st[10000],s[10000]; //st用来接收输入的字符串，s用来接收解压的字符串

int cn[257];

HuffmanTree HT;

HuffmanCode HC;

char ch='y';

int d,i;

printf("---------------霍夫曼树-------------\n\n");

printf(" 1.建立霍夫曼树.\n");

printf(" 2.生成霍夫曼编码.\n");

printf(" 3.编码文件译码.\n");

while(ch=='y')

{

printf("请选择：");

scanf("%d",&d);

switch(d)

{

case 1:

printf("输入要编码的字符串：");

scanf("%s",&st);

num=jsq(st,cn,str); //统计文件中的字符

str[num+1]='\0';

chuffmanTree(HT,HC,cn,str);

printf("霍夫曼树建立成功！\n");//建立霍夫曼树

break;

case 2:

HuffmanEncoding(HT,HC); //根据霍夫曼树建立霍夫曼编码

printf("建立霍夫曼编码如下\n 共有%d种字符\n",num);

for(i=1;i<=num;i++)

printf("字符%c次数为%d,编码为%s\n",HC[i].ch,cn[i],HC[i].bits);

char get[10000];

get[0]='\0';

coding(HC,st,get);

printf("\n上述字符串的霍夫曼编码为%s\n",get);

break;

case 3:

printf("输入二进制码开始译码：");

char receive[10000];

scanf("%s",receive);

coding(HC,receive,s); //译码

printf("译码为：%s\n",s);

break;

}

printf("\n是否继续？(y/n):");

scanf("%c",&ch);

scanf("%c",&ch);

}

return 0;

}

**运行结果：**

