**实 验 报 告**

**课程名称** 数据结构

**实验项目**  实验二 哈夫曼编码及译码

**系 别\_\_\_ \_计算机学院 \_ \_\_\_\_\_\_**

**专 业\_\_\_\_计算机科学与技术\_ \_\_**

**班级/学号\_\_\_\_1503/2015011227 \_\_**

**学生姓名 \_\_\_\_\_\_\_徐蕾 \_\_\_\_\_\_\_\_\_\_ \_**

**实验日期** \_ 2017年3月21日

**成 绩** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_

**指导教师** 黄改娟

**实验题目：实验二 哈夫曼编码及译码**

1. **实验目的**

设计一个实现哈夫曼编码及译码的演示程序

1. **实验内容**

利用哈夫曼树实现编码及译码

1. **报告内容**

程序源码：

#include <stdio.h>

#include <stdlib.h> //要用system函数要调用的头文件

#include<conio.h> //用getch()要调用的头文件

#include <string.h>

#define N 50 //义用N表示50叶节点数

#define M 2\*N-1 //用M表示节点总数 当叶节点数位n时总节点数为2n-1

#define MAXSIZE 100

typedef struct

{

char data; //结点值

int weight; //权值

int parent; //双亲结点

int lchild; //左孩子结点

int rchild; //右孩子结点

}HTNode;

typedef struct

{

char cd[N]; //存放哈夫曼码

int start; //从start开始读cd中的哈夫曼码

}HCode;

void CreateHT(HTNode ht[],int n) //调用输入的数组ht[],和节点数n

{

int i,k,lnode,rnode;

int min1,min2;

for (i=0;i<2\*n-1;i++)

ht[i].parent=ht[i].lchild=ht[i].rchild=-1; //所有结点的相关域置初值-1

for (i=n;i<2\*n-1;i++) //构造哈夫曼树

{

min1=min2=32767; //int的范围是-32768—32767

lnode=rnode=-1; //lnode和rnode记录最小权值的两个结点位置

for (k=0;k<=i-1;k++)

{

if (ht[k].parent==-1) //只在尚未构造二叉树的结点中查找

{

if (ht[k].weight<min1) //若权值小于最小的左节点的权值

{

min2=min1;rnode=lnode;

min1=ht[k].weight;lnode=k;

}

else if (ht[k].weight<min2)

{

min2=ht[k].weight;rnode=k;

}

}

}

ht[lnode].parent=i;ht[rnode].parent=i; //两个最小节点的父节点是i

ht[i].weight=ht[lnode].weight+ht[rnode].weight; //两个最小节点的父节点权值为两个最小节点权值之和

ht[i].lchild=lnode;ht[i].rchild=rnode; //父节点的左节点和右节点

}

}

void CreateHCode(HTNode ht[],HCode hcd[],int n)

{

int i,f,c;

HCode hc;

for (i=0;i<n;i++) //根据哈夫曼树求哈夫曼编码

{

hc.start=n;c=i;

f=ht[i].parent;

while (f!=-1) //循序直到树根结点结束循环

{

if (ht[f].lchild==c) //处理左孩子结点

hc.cd[hc.start--]='0';

else //处理右孩子结点

hc.cd[hc.start--]='1';

c=f;f=ht[f].parent;

}

hc.start++; //start指向哈夫曼编码hc.cd[]中最开始字符

hcd[i]=hc;

}

}

void DispHCode(HTNode ht[],HCode hcd[],int n) //输出哈夫曼编码的列表

{

int i,k;

printf(" 输出哈夫曼编码:\n");

for (i=0;i<n;i++) //输出data中的所有数据，即A-Z

{

printf(" %c:\t",ht[i].data);

for (k=hcd[i].start;k<=n;k++) //输出所有data中数据的编码

{

printf("%c",hcd[i].cd[k]);

}

printf("\n");

}

}

void editHCode(HTNode ht[],HCode hcd[],int n) //编码函数

{

char string[MAXSIZE];

int i,j,k;

scanf("%s",string); //把要进行编码的字符串存入string数组中

printf("\n输出编码结果:\n");

for (i=0;string[i]!='#';i++) //#为终止标志

{

for (j=0;j<n;j++)

{

if(string[i]==ht[j].data) //循环查找与输入字符相同的编号，相同的就输出这个字符的编码

{

for (k=hcd[j].start;k<=n;k++)

{

printf("%c",hcd[j].cd[k]);

}

break; //输出完成后跳出当前for循环

}

}

}

}

void deHCode(HTNode ht[],HCode hcd[],int n) //译码函数

{

char code[MAXSIZE];

int i,j,l,k,m,x;

scanf("%s",code); //把要进行译码的字符串存入code数组中

while(code[0]!='#')

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

{

m=0; //m为相同编码个数的计数器

for (k=hcd[i].start,j=0;k<=n;k++,j++) //j为记录所存储这个字符的编码个数

{

if(code[j]==hcd[i].cd[k]) //当有相同编码时m值加1

m++;

}

if(m==j) //当输入的字符串与所存储的编码字符串个数相等时则输出这个的data数据

{

printf("%c",ht[i].data);

for(x=0;code[x-1]!='#';x++) //把已经使用过的code数组里的字符串删除

{

code[x]=code[x+j];

}

}

}

}

int main()

{

int n=26,i;

char orz,back,flag=1;

char str[]={'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'}; //初始化

int fnum[]={186,64,13,22,32,103,21,15,47,57,1,2,32,20,57,63,15,1,48,51,80,23,8,18,1,16}; //初始化

HTNode ht[M]; //建立结构体

HCode hcd[N]; //建立结构体

for (i=0;i<n;i++) //把初始化的数据存入ht结构体中

{

ht[i].data=str[i];

ht[i].weight=fnum[i];

}

while (flag) //菜单函数，当flag为0时跳出循环

{

printf("\n");

printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n \*\* A---------------显示编码 \*\*");

printf("\n \*\* B---------------进行编码 \*\*");

printf("\n \*\* C---------------进行译码 \*\*");

printf("\n \*\* D---------------退出 \*\*\n");

printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n");

printf(" 请输入选择的编号:");

scanf("%c",&orz);

switch(orz)

{

case 'a':

case 'A':

system("cls"); //清屏函数

CreateHT(ht,n);

CreateHCode(ht,hcd,n);

DispHCode(ht,hcd,n);

printf("\n按任意键返回...");

getch();

system("cls");

break;

case 'b':

case 'B':

system("cls");

printf("请输入要进行编码的字符串(以#结束):\n");

editHCode(ht,hcd,n);

printf("\n按任意键返回...");

getch();

system("cls");

break;

case 'c':

case 'C':

system("cls");

DispHCode(ht,hcd,n);

printf("请输入编码(以#结束):\n");

deHCode(ht,hcd,n);

printf("\n按任意键返回...");

getch();

system("cls");

break;

case 'd':

case 'D':

flag=0;

break;

default:

system("cls");

}

}

}

运行结果：







