哈夫曼编/译码器

电信1808 谢学泷 U201890044 完成日期 4.25

1. 需求分析

输入一段字符串

将其编码并输出，再根据编码进行翻译，输出翻译的字符串

功能：实现哈夫曼编码和译码

测试数据：

输入

THIS PROGRAM IS MY FAVORITE

输出

0010

1110

1001

1100

000

011101

1101

0110

011110

1101

0101

001101

000

1001

1100

000

001101

011100

000

001100

0101

0011111

0110

1101

1001

0010

101

THIS RPOGRAM IS MY FAVORITE

1. 概要设计

本题使用结构体数组来存储哈夫曼树

typedef struct nodelist{

int f; // 节点的频率

char data; // 节点代表的字符

int lchild; // 节点的左右子节点

int rchild;

}tree;

void Initialization(int a[],char b[],int n,tree hfm[])//建立哈夫曼树

void makeit(int a[],char c[][27],int q,tree hfm[]) //根据递归对已生成的哈夫曼树进行编码(哈夫曼树的遍历)

char translate(char s[],tree hfm[],int q) //根据编码进行翻译

main函数调用其他函数完成程序。

3.详细设计

建立哈夫曼树

void Initialization(int a[],char b[],int n,tree hfm[])

{

int pot[100],i,m,mm,s=n,min,mmin,t; // pot标记为加入树的节点 m,mm,min,mmin代表最小值和次小值及其节点的编号

for (i=0;i<100;i++) pot[i]=0; //初始化

for (i=0;i<n;i++) {hfm[i].data=b[i]; hfm[i].f=a[i]; pot[i]=1; hfm[i].lchild=-1; hfm[i].rchild=-1;}

while (s<2\*n-1) // 哈夫曼树为完全二叉树，其节点数为2n-1

{

min=1000;mmin=1001;

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

{

if (pot[i]!=0&&hfm[i].f<mmin) {mmin=hfm[i].f; mm=i;}

if (mmin<min) {t=min; min=mmin; mmin=t; t=mm; mm=m; m=t;}

} // 找到最小值和次小值

hfm[s].lchild=mm; hfm[s].rchild=m;

hfm[s].f=min+mmin;

pot[m]=0; pot[mm]=0; pot[s]=1; s++; //进行操作

}

return;

}

4.调试分析

时间复杂度 O（n^2）

空间复杂度O(n^2)

5.用户使用说明

修改S串的值运行

6.测试结果

S串为XIEXUELONG

输出结果：0011110101

1001

101

0011110101

11110

101

01000

0110

1000

011110

XIEXUELONG