哈夫曼树

电信1808 连琛 U201813455 2019.4.22

利用哈夫曼编码进行通信可以大大提高信道利用率，缩短信息传输时间，降低传输成本。但是，这要求在发送端通过一个编码系统对待传数据预先编码，在接收端将传来的数据进行译码（复原）。对于双工信道（即可以双向传输信息的信道），每端都需要一个完整的编/译码系统。试为这样的信息收发站写一个哈夫曼码的编/译系统。

1. 需求分析

任务要求输入一段字符串，将其编码输出，在根据编码进行翻译，将所译字符串输出。

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。

1. 详细设计

建立哈夫曼树

for (i=0;i<100;i++) pot[i]=0; //初始化 pot为标记数组

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++;

}

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)

{ int r;

if (hfm[q].lchild>=0) {r=hfm[q].lchild; strcpy(c[r],c[q]); c[r][strlen(c[r])]='0'; makeit(a,c,r,hfm);}

if (hfm[q].rchild>=0) {r=hfm[q].rchild; strcpy(c[r],c[q]); c[r][strlen(c[r])]='1'; makeit(a,c,r,hfm);}

}

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

{

int i=0,j=q;

while (i<strlen(s))

{

if (s[i]=='0') j=hfm[j].lchild;

else j=hfm[j].rchild;

i++;

}

return hfm[j].data;

}

1. 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

在程序中修改S串的值运行即可。

6.测试结果

输入

HELLO WORLD

输出

1110

101

01000

01000

0110

000

001110

0110

1101

01000

01001

HELLO WORLD