题目：哈夫曼编/译码器

班级：电信1808

学号：U201813470

完成日期：2019.04.25

1. 分析思路

将频度作为权重，每次将频度最小的两个作为子节点，新开一个节点作为根节点，根节点权重为子节点权重之和，由此建立哈夫曼树。编码过程，从叶子到根逆向求编码，双亲与左孩子间编码为0，与右孩子间编码为1。

1. 具体操作

1.建立节点的结构体

2.建立长度为m=2\*n-1的一维数组，前n个为叶子节点，后n-1个为双亲结点。

3.将最小的两个节点指向从n+1开始的同一个根节点，直到n个叶子节点都有双亲。

4.从数组中第一个叶子节点开始逆向编码，存到新开的指针数组指向的空间中。

三．主要函数

viod HuffmanCoding(HuffmanTree HT,HumanCOde HC,int \*w,int n)

{

if(n<=1) return;

m=2\*n-1;

HTNode \*p;

HT=(HuffmanTree)malloc((m+1)\*sizeof(HTNode));

for(p=HT+1,i=1;i<=n;++i,++p,++w)

{

p->weight=\*w;

p->lchild=0;

p->parent=0;

p->rchild=0;

}

for(i=n+1;i<=m;i++)

{

p->lchild=0;

p->parent=0;

p->rchild=0;

p->weight=0;

}

for(i=n+1;i<=m;++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;

}

char \*\* HC=(HuffmanCode)malloc((n+1)\*sizeof(char \*));

char \* cd=(char \*)malloc(n\*sizeof(char));

cd[n-1]="\0";

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

{

start=n-1;

for(int c=i,int f=HT[i].parent;f!=0;c=f,f=HT[f].parent)

{

if(HT[f].lchild==c) cd[--start]="0";

else cd[--start]="1";

}

HC[i]=(char\*)malloc((n-start)\*sizeof(char));

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

}

free(cd);

}

四．收获

学会了哈夫曼树的构建以及编码的实现。