# 第四次上机实验报告

姓名：康辉 学号：15180210013 班级：1518021

1. 题目要求

对输入的一串电文字符(A B C D E F G H 8 个字符，其概率分别为 0.05, 0.29, 0.07, 0.08, 0.14, 0.23, 0.03, 0.11)实现Huffman编码，再对Huffman编码生成的代码串进行译码，输出电文字符串。实现功能如下：

• Huffman树的建立

• Huffman编码的生成

• 编码文件的译码

1. 程序思路

设置一个数组v存放待编码元素，一个数组w存放权值，n为元素个数，ht为创建的哈夫曼树，hc为指针数组存储哈夫曼编码，这些作为函数参数传递到函数haffcoding中，创建编码树，然后按权值进行编码，这些都在一个函数中实现。

哈夫曼树节点：

typedef struct

{ char data;

Int weight;

Int parent,lchild,rchild;

}htnode,huffmantree;

1. 程序设计中遇到的问题
2. 选择一个数组来存储来存放编码，但是指针数组运用的不够熟练，造成值传递时出错。最后查看c语言书籍得以解决.
3. 在选择所有权值中最小，次小的时候出现错误，造成乱序，编出来的编码不正确，最后通过单步调试找出错误。
4. hc=(huffmancode)malloc((n+1)\*sizeof(char \*));这里第一次分配空间为char型，对指针数组很不熟悉造成的。
5. 还有其他的很多问题在程序中都有注释加重点。
6. 程序源码

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#define length sizeof(htnode)

typedef struct htnode

{

char data;

int weight;

int parent,lchild,rchild;

}htnode,\*huffmantree; //哈夫曼存储结构

typedef char \*\*huffmancode; //存储哈夫曼编码表

void huffmancoding(huffmantree ht,huffmancode &hc,int \*w,char \*v,int n)//v是待编码元素数组，w是每个元素的权值，构造哈夫曼树ht，hc存放哈夫曼编码

{

int i,m,s1,s2,min1,min2,j;

char \*cd;

int start,c,f;

if(n<=1) return;

m=2\*n-1;

ht=(huffmantree)malloc((m+1)\*length); //有n个叶子节点的hfuumantree需要2n-1个结点存储

for(i=1;i<=n;++i,++w,++v) //对n个元素进赋值，权值，父亲左右孩子标记为0

{

ht[i].data=\*v;ht[i].weight=\*w;ht[i].lchild=0;ht[i].parent=0;ht[i].rchild=0;

}

for(;i<=m;++i) //对后m-n个非叶子节点标记

{

ht[i].data=0;ht[i].weight=0;ht[i].lchild=0;ht[i].parent=0;ht[i].rchild=0 ;

}

for(i=n+1;i<=m;++i) //建树

{

s1=s2=0;min1=min2=32767;

for(j=1;j<=i-1;++j)

{

if(ht[j].parent==0)

{if(ht[j].weight<min1)

{min2=min1;min1=ht[j].weight;s2=s1;s1=j;} //此第一次处排序出错

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

{min2=ht[j].weight;s2=j;}}

}

ht[s1].parent=i;ht[s2].parent=i;

ht[i].lchild=s1;ht[i].rchild=s2;

ht[i].weight=min1+min2;

}

//一下为从叶子结点到根节点求哈夫曼编码

hc=(huffmancode)malloc((n+1)\*sizeof(char \*)); //分配n个字符编码的头指针向量

cd=(char\*)malloc(n\*sizeof(char)); //分配求编码的工作空间

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

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

{

start=n-1;

for(c=i,f=ht[i].parent;f!=0;c=f,f=ht[f].parent) //叶子到根

{

if(ht[f].lchild==c) //////=======

cd[--start]='0';

else

cd[--start]='1';

}

hc[i]=(char \*)malloc((n-start)\*sizeof(char)); //为第i个字符编码分配空间

strcpy(hc[i],cd+start);

}

free(cd);

} //huffmancoding

int main()

{

printf("\*\*\*\*\*\*\*\*\*\*哈夫曼编码\*\*\*\*\*\*\*\*\*\*\* by-KangHui\n");

int n,i;

char \*v; //存字符

int \*w; //权值

huffmantree ht;

huffmancode hc;

printf("请输入需要编码元素的个数:");

scanf("%d",&n);

printf("请输入要编码的字符串(连续输入不要有空格:)") ;

v=(char \*)malloc((n+2)\*sizeof(char)) ;

v[n+1]='\0';

scanf("%s",v);

w=(int \*)malloc((n+1)\*sizeof(int));

printf("请输入每个字符的权值(空格隔开):");

for(i=0;i<n;++i) //i=1错误，可能会造成传输过去形参空执政

{

scanf("%d",w+i);

}

huffmancoding(ht,hc,w,v,n);

printf("各个字符哈夫曼编码分别为:");

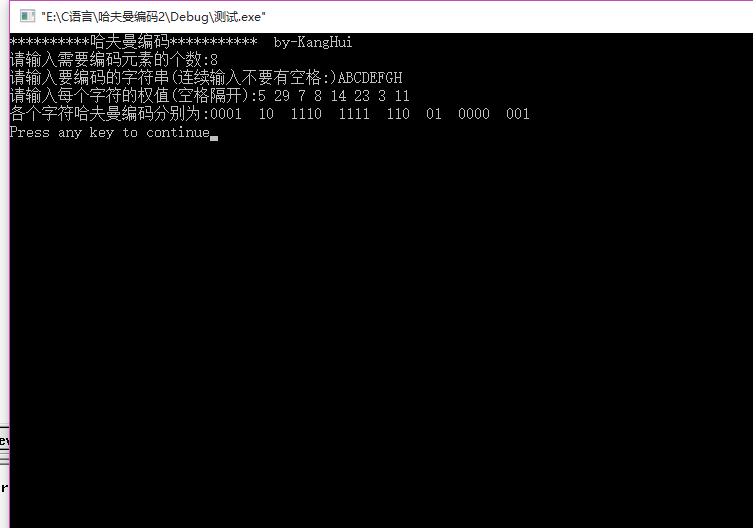
for(i=1;i<=n;i++)

printf("%s ",hc[i]); ///这里竟然他妈一直用的ht，让我找了那么久= =

printf("\n");

return 0;

}

运行截图：

1. 编程总结

最大的感受就是单步调试在程序修改中的重要性，以前写的程序都比较少，有错误直接就可以看出来，现在程序比较长，不易查错。我第一次在打印编码的时候不小心将hc写成ht，造成程序运行崩溃，但是并没有提示语法错误。还有赋值权值的时候没有从开辟的第一个空间开始，造成第一个空间为空。这几个错误我找了将近一两个小时。当不会使用单步调试的时候出现逻辑错误是非常难以找出来的。所以我一定要先学会单步调试程序。