数据结构实验报告

院系：信息工程学院

专业：软件工程二班

学号：20122480212

姓名：霍静静

赫夫曼编码的实验报告

**一、实验目的**：了解赫夫曼树的存储结构，熟练掌握赫夫曼算法；

**二、实验要求**：

1）定义赫夫曼树的存储结构；

2）熟练掌握赫夫曼编码的思想和算法；

**三、实验内容：**

**读入各个符号及其权值，求各符号的哈夫曼编码。**

**四、算法思想：**

1.通过赫夫曼树的建立完成赫夫曼编码的生成；

2.构造赫夫曼树：1） 将给定的n个权值{w1,w2,…,wn}作为n个根结点的权值构造一个具有n棵二叉树的 森林{T1,T2,…,Tn}，其中每棵树只有一个根结点； （2） 在森林中选取两棵根结点权值最小的二叉树作为左右子树，构造一棵新二叉树，新 二叉树新结点的权值为这两棵树根的权值之和； （3） 在森林中，将上面选择的这两棵权值最小的二叉树从森林中删除，并将刚刚新构造 的二叉树加入到森林中； （4） 重复上面的（2）和（3），知道森林中只有一棵二叉树为止。这棵二叉树就是赫夫曼树。

**五、主要函数的实现：**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<string.h>**

**typedef struct{**

**unsigned int weight;**

**int parent,lchild,rchild;**

**}HTNode,\*HuffmanTree; //动态分配数组存储赫夫曼树**

**typedef char \*\*HuffmanCode;//动态分配数组存储赫夫曼编码表**

**void Select(HuffmanTree HT,int n,int &s1,int &s2)**

**{**

**unsigned int m1,m2;**

**int k;**

**m1=m2=32767;**

**for(k=1;k<=n;k++)**

**{**

**if((HT[k].parent==0)&&(HT[k].weight<m1))**

**{**

**m2=m1;s2=s1;**

**m1=HT[k].weight;**

**s1=k;**

**}**

**else if((HT[k].parent==0)&&(HT[k].weight<m2))**

**{**

**m2=HT[k].weight;**

**s2=k;**

**}**

**}**

**}**

**void HuffmanCoding(HuffmanTree &HT,HuffmanCode &HC,unsigned int \*w,int n)**

**{**

**int i,m,start,c,f;**

**int s1=0,s2=0;**

**char \*cd;**

**HuffmanTree p;**

**if(n<=1) return;**

**m=2\*n-1;**

**HT=(HuffmanTree)malloc((m+1)\*sizeof(HTNode));//0号单元未用**

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

**{**

**p->weight=\*w;**

**p->parent=0;**

**p->lchild=0;**

**p->rchild=0;**

**}**

**for(;i<=m;++i,++p)**

**{**

**p->weight=0;**

**p->parent=0;**

**p->lchild=0;**

**p->rchild=0;**

**}**

**for(i=n+1;i<=m;++i)//建赫夫曼树**

**{//----在HT[1...i-1]选择parent为0且weight最小的两个接点，其序号分别为s1,s2----**

**Select(HT,i-1,s1,s2);**

**HT[s1].parent=i;HT[s2].parent=i;**

**if(s1<s2)**

**{HT[i].lchild=s1;HT[i].rchild=s2;}**

**else**

**{HT[i].lchild=s2;HT[i].rchild=s1;}**

**HT[i].weight=HT[s1].weight+HT[s2].weight;**

**}**

**//----从叶子到根逆向求每个字符的赫夫曼编码----**

**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'; //编码结束符**

**for(i=1;i<=n;++i) //逐个字符求赫夫曼编码**

**{**

**start=n-1; //编码结束符位置**

**for(c=i,f=HT[i].parent;f!=0;c=f,f=HT[f].parent)//从叶子到根逆向求编码**

**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]); //从cd复制编码到HC**

**}**

**free(cd);**

**}**

**int main()**

**{**

**unsigned int \*weigh;**

**int i,n;**

**HuffmanTree ht;**

**HuffmanCode hc;**

**printf("请输入权值个数：");**

**scanf("%d",&n);**

**weigh=(unsigned int \*)malloc(n\*sizeof(unsigned int));**

**printf("请输入%d个权值：",n);**

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

**scanf("%u",weigh+i);**

**HuffmanCoding(ht,hc,weigh,n);**

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

**{printf("%u\t%d\t%d\t%d\n",ht[i].weight,ht[i].parent,ht[i].lchild,ht[i].rchild);}**

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

**{**

**printf("%u:",ht[i].weight);**

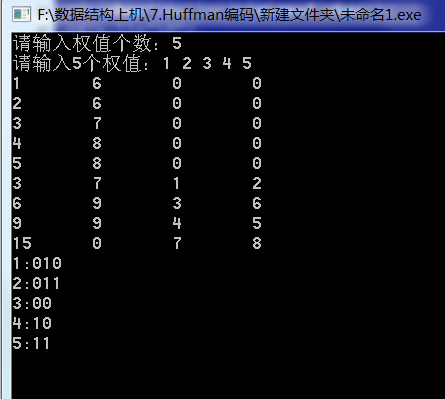
**puts(hc[i]);**

**}**

**getchar(); getchar(); getchar(); getchar(); getchar();**

**}**

**六、测试结果：**



**七、实验总结：**

**在编写程序的过程中，遇到一些字符错误，缺少符号是在所难免的，可以通过认真检查看出。然后就是核心函数，赫夫曼编码算法的编写，无法脱离课本。**