

**教学上机实验报告**

**课程名称：**   **数据结构**

**任课教师姓名：**  **贾盼盼**

**学生学号：**  312105010207

**学生姓名：**  刘晨阳

**学生专业班级：** 计算2106

**2021 ～ 2022 学年 第一学期**

|  |
| --- |
| **河南理工大学**  **教学上机实验报告评价分值标准** |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 序号 | 评价指标 | 分值 | 评价等级及参考分值 | | | | | 评价分 | | 优 | 良 | 中 | 合格 | 差 | | 1 | 实验报告内容完整充实 | 10 | 10 | 8 | 7 | 6 | 3 |  | | 2 | 实验内容书写规范、字迹工整认真 | 10 | 10 | 8 | 7 | 6 | 3 |  | | 3 | 实验过程叙述详细、概念正确，语言表达准确，结构严谨，调理清楚，逻辑性强，自己努力完成，没有抄袭。 | 30 | 30 | 26 | 23 | 20 | 10 |  | | 4 | 对实验过程中存在的问题分析详细透彻、深刻、全面、规范、，结合实验内容，有自己的个人见解和想法，并能结合该实验提出相关问题，给出解决方法。 | 30 | 30 | 26 | 23 | 20 | 10 |  | | 5 | 实验结果、分析和结论正确无误 | 20 | 20 | 17 | 15 | 13 | 6 |  | | 总得分 | | | | | | | |  |   签名（签章）：  日期： 年 月 日 |
|  |

|  |
| --- |
| **河南理工大学教学上机实验报告** |
| 上机时间 2022 年 11月20 日 |
| **实验题目：**  设计实现Huffman树的创建算法，并利用Huffman树进行编码和译码。 |
| **实验目的和要求：**  **实验目的：**熟练掌握二叉树的基本操作操作，应用二叉树的结构解决实际问题，熟练掌握Huffman树和Huffman编码的构造方法。  **实验要求**： 1.在“Huffman编码“基础上的实现译码操作如输入“10010”译出相应的字符。 2.选做：(1)增加统计原文中各字符权值的功能,(2)将原文进行编码。 |
| **实验过程：**  **程序主要功能源代码：**  **//算法5.11 根据赫夫曼树求赫夫曼编码**  **#include<iostream>**  **#include<cstring>**  **using namespace std;**  **typedef struct**  **{**  **int weight;**  **int parent,lchild,rchild;**  **}HTNode,\*HuffmanTree;**  **typedef char \*\*HuffmanCode;**  **void Select(HuffmanTree HT,int len,int &s1,int &s2)**  **{**  **int i,min1=INT\_MAX,min2=INT\_MAX;//先赋予最大值**  **for(i=1;i<=len;i++)**  **{**  **if(HT[i].weight<min1&&HT[i].parent==0)**  **{**  **min1=HT[i].weight;**  **s1=i;**  **}**  **}**  **int temp=HT[s1].weight;//将原值存放起来，然后先赋予最大值，防止s1被重复选择**  **HT[s1].weight=INT\_MAX;**  **for(i=1;i<=len;i++)**  **{**  **if(HT[i].weight<min2&&HT[i].parent==0)**  **{**  **min2=HT[i].weight;**  **s2=i;**  **}**  **}**  **HT[s1].weight=temp;//恢复原来的值**  **}**  **//用算法5.10构造赫夫曼树**  **void CreatHuffmanTree(HuffmanTree &HT,int n)**  **{**  **//构造赫夫曼树HT**  **int m,s1,s2,i;**  **if(n<=1) return;**  **m=2\*n-1;**  **HT=new HTNode[m+1]; //0号单元未用，所以需要动态分配m+1个单元，HT[m]表示根结点**  **for(i=1;i<=m;++i) //将1~m号单元中的双亲、左孩子，右孩子的下标都初始化为0**  **{ HT[i].parent=0; HT[i].lchild=0; HT[i].rchild=0; }**    **cout<<"请输入叶子结点的权值：\n";**  **for(i=1;i<=n;++i) //输入前n个单元中叶子结点的权值**  **cin>>HT[i].weight;**  **/\*――――――――――初始化工作结束，下面开始创建赫夫曼树――――――――――\*/**  **for(i=n+1;i<=m;++i)**  **{ //通过n-1次的选择、删除、合并来创建赫夫曼树**  **Select(HT,i-1,s1,s2);**  **//在HT[k](1≤k≤i-1)中选择两个其双亲域为0且权值最小的结点,**  **// 并返回它们在HT中的序号s1和s2**  **HT[s1].parent=i;**  **HT[s2].parent=i;**  **//得到新结点i，从森林中删除s1，s2，将s1和s2的双亲域由0改为i**  **HT[i].lchild=s1;**  **HT[i].rchild=s2 ; //s1,s2分别作为i的左右孩子**  **HT[i].weight=HT[s1].weight+HT[s2].weight; //i 的权值为左右孩子权值之和**  **} //for**  **}**  **// CreatHuffmanTree**  **void CreatHuffmanCode(HuffmanTree HT,HuffmanCode &HC,int n)**  **{**  **//从叶子到根逆向求每个字符的赫夫曼编码，存储在编码表HC中**  **int i,start,c,f;**  **HC=new char\*[n+1]; //分配n个字符编码的头指针矢量**  **char \*cd=new char[n]; //分配临时存放编码的动态数组空间**  **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; //start开始时指向最后，即编码结束符位置**  **c=i;**  **f=HT[i].parent; //f指向结点c的双亲结点**  **while(f!=0)**  **{ //从叶子结点开始向上回溯，直到根结点**  **--start; //回溯一次start向前指一个位置**  **if(HT[f].lchild==c)**  **cd[start]='0'; //结点c是f的左孩子，则生成代码0**  **else**  **cd[start]='1'; //结点c是f的右孩子，则生成代码1**  **c=f;**  **f=HT[f].parent; //继续向上回溯**  **} //求出第i个字符的编码**  **HC[i]=new char[n-start]; // 为第i 个字符编码分配空间**  **strcpy(HC[i], &cd[start]); //将求得的编码从临时空间cd复制到HC的当前行中**  **}**  **delete cd; //释放临时空间**  **} // CreatHuffanCode**  **void show(HuffmanTree HT,HuffmanCode HC,int n)**  **{**  **for(int i=1;i<=n;i++)**  **cout<<HT[i].weight<<"编码为"<<HC[i]<<endl;**  **}**  **/\*void decode(HuffmanTree &HT,char \*ch,int n)//依次读入电文，根据哈夫曼树译码**  **{**  **int i,j=0;**  **char b[100];**  **char endflag='#'; //电文结束标志取#**  **i=2\*n-1; //从根结点开始往下搜索**  **getchar();**  **cout<<("输入发送的编码(以'#'为结束标志)：");**  **gets(b);**  **cout<<"译码后的字符为";**  **while(b[j] != endflag)**  **{**  **//填写完整**  **}**  **cout<<endl;**    **}//decode**  **\*/**  **int main()**  **{**  **HuffmanTree HT;**  **HuffmanCode HC;**  **int n;**  **cout<<"请输入待编码的字符(叶子结点)的个数：\n";**  **cin>>n;**  **getchar();**  **char \*p=new char[n];//?**  **printf("请输入待编码的字符(中间不加空格)");**  **for(int i=1;i<=n;i++)**  **cin>>p[i];**  **CreatHuffmanTree(HT,n);**  **CreatHuffmanCode(HT,HC,n);**  **show(HT,HC,n);**  **//decode(HT,p,n);**  **return 1;**  **}** |
| **实验结果：** |
| **实验分析：**  首先构造两个结构体分别存储结点的字符及权值、哈夫曼编码值，然后读取前n个结点的字符及权值，建立哈夫曼树，读取前n个结点的字符及权值，建立哈夫曼树，读入电文，最后根据哈夫曼树译码。本次实验更好的了解了哈夫曼夫的构造与算法 |
| **实验成绩：**  日期： 年 月 日 |