《数据结构》课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业名称 | 计算机 | 年级 | 2017 | 班级 | 2 |
| 学生姓名 | 王汝芸 | 学号 | 201711010202 | 指导教师 | 郑志华 |
| 实验题目 | 实验5 创建哈夫曼树 | | | 提交时间 | 2018.5.2 |

一、实验目的和要求

（1）创建哈夫曼树

（2）输出哈夫曼树每个节点的权重、父亲、左右儿子

二、实验环境

Visual Studio2017

三、实验内容及实施

定义一个struct储存节点，再用此类型开辟一个数组，依次输入输出

**实验5：哈夫曼树的创建**

实验要求：

1、从键盘输入n, 以及n个字符的概率。

2、用顺序存储。

**模块图**

建立2\*n-1个大小的数组

查找最小元素和次小元素

为新节点赋值

为叶子结点赋值

打印结果

1、从键盘输入n, 以及n个字符的概率。

2、用顺序存储。

**【源程序】**

/\*创建哈夫曼树\*/

/\*

1、从键盘输入n, 以及n个字符的概率。

2、用顺序存储

\*/

#include<stdio.h>

#include<stdlib.h>

#define N 8

typedef struct HuffmanTreeNode

{

int power;

int parent;

int Lchild;

int Rchild;

}HuffTreeNode;

void minPower(HuffmanTreeNode \* a)

{

int min1 = 100, min2 =100;

int i, j, k;

int mini,minj;

int count = N;//当前总节点数

for (k = 0; k < count; k++)

{

//寻找最小值和次小值

for (i = 0; i < count; i++)//找最小值

{

if ((!a[i].parent) && (a[i].power <= min1))

{

min1 = a[i].power;

mini = i;

}

}

for (j = 0; j < count; j++)//找次小值

{

if ((!a[j].parent) && (a[j].power <= min2) && (a[j].power != min1))

{

min2 = a[j].power;

minj = j;

}

}

//叶子结点赋值父亲

a[mini].parent = count+1;

a[minj].parent = count+1;

//新节点赋值power和儿子

a[count].power = min1 + min2;

a[count].Lchild = mini + 1;

a[count].Rchild = minj + 1;

count++;

min1 = 100;min2 = 100;

if (count == 2 \* N - 1)

{

break;

}

}

for (i = 0; i < 2 \* N - 1; i++)//打印结果

{

printf("第%d个节点的权是：%d，Parent = %d，Lchild = %d，Rchild = %d\n", i + 1, a[i].power, a[i].parent, a[i].Lchild, a[i].Rchild);

}

}

int main()

{

printf("\n----该程序用于创建%d个叶子节点的哈夫曼树----\n\n", N);

HuffmanTreeNode a[2 \* N-1];

//初始化

for (int i = 0; i < 2 \* N - 1; i++)

{

a[i].parent = a[i].power = a[i].Lchild = a[i].Rchild = 0;

}

printf("请输入%d个叶子结点的权值：\n",N);

//录入数据

for (int i = 0; i < N; i++)

{

scanf\_s("%d", &a[i].power);

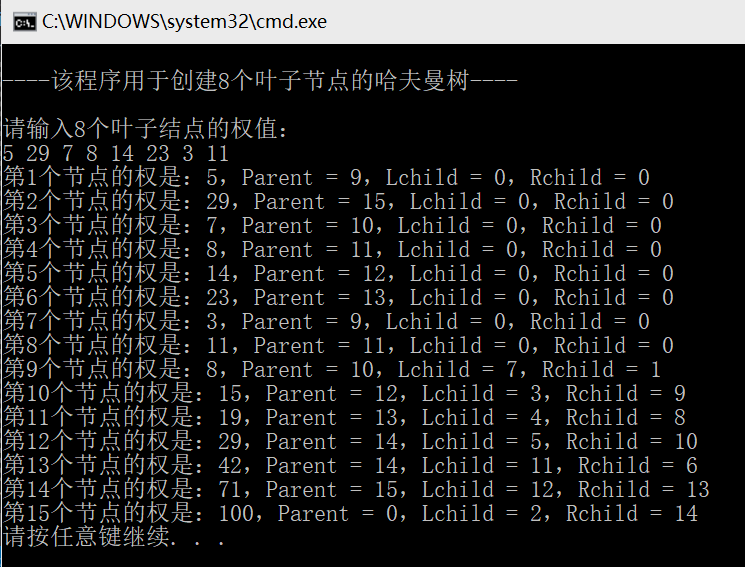
}

minPower(a);

return 0;

}

**四、实验结果 (程序的执行结果)**



**五、实验讨论（可选）**

无