《离散数学》课程实验报告5 最优二元树

**1、题目简介**

题目简介：输入一组通信符号的使用频率，求各通信符号对应的前缀码。

要求：（1）用一维数组f[N]存贮通信符号的使用频率，用求最优2元树的方法求出每个通信符号的前缀码。

（2）用链表保存最优2元树，输出前缀码时可以用树的遍历方法。

**2、解题思路**

由给定的 n 个权值，构造具有 n 棵扩充二叉树的森林 棵扩充二叉树的森林F = { T0, T1,T2,

…, Tn-1 }，其中每一棵扩充二叉树 其中每一棵扩充二叉树Ti 只有一个带有权值 只有一个带有权值wi 的根结点，其左、右子树均为空。 的根结点，其左、右子树均为空。

(2) 重复以下步骤, 直到 F 中仅剩下一棵树为止 中仅剩下一棵树为止：

①在 F 中选取两棵根结点的权值最小的扩充二 中选取两棵根结点的权值最小的扩充二叉树, 做为左、右子树构造一棵新的二叉树。置新 做为左、右子树构造一棵新的二叉树。置新

的二叉树的根结点的权值为其左、右子树上根结 的二叉树的根结点的权值为其左、右子树上根结点的权值之和。 点的权值之和。

②在 F 中删去这两棵二叉树。

最后得到的结点就是霍夫曼树的根结点，随后只需要定义一个char数组，中序遍历一遍二叉树，在左子树上输出0，在右子树上输出1，最后得到不同的前缀码。

**3、数据结构设计**

这个作业的核心数据结构是二叉树，所以我设计了一个树类，以及树的结点类，在树类中定义了各种方法构造霍夫曼树。

**4、核心算法**

**构造霍夫曼树**

void Tree :: construct\_tree()

{

for (int n = 1; n < leafnum; n++)

{

int min = leafnum-n;

int submin =leafnum-n-1 ;

Node\* newnode = new Node;

newnode->num = leaf[min]->num + leaf[submin]->num;

newnode->Lnode = leaf[min];

newnode->Rnode = leaf[submin];

//插入新叶子结点

int position = 0;

while (leaf[position]->num >= newnode->num)

{

position++;

}

while (min > position)

{

leaf[min] = leaf[min-1];

min--;

}

leaf[position] = newnode;

}

root = leaf[0];

}

//输出霍夫曼编码

void Tree::preorder()

{

key[keynum] = '0';

keynum++;

Preorder(root,1);

}

void Tree::Preorder(Node\* node,int depth)

{

if (node->Lnode == NULL && node->Rnode == NULL)//如果是叶子了，输出

{

cout << node->num << ": ";

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

cout << key[i];

cout << endl;

return;

}

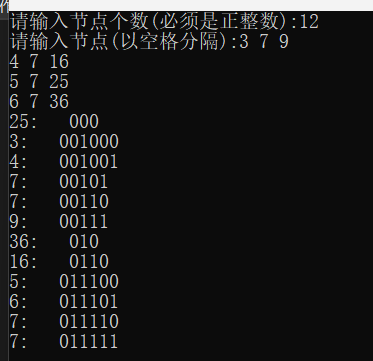
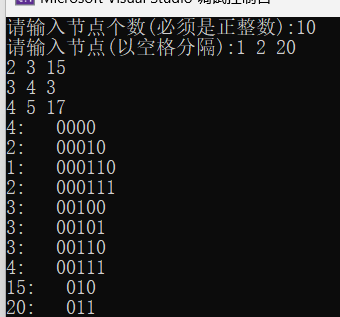
key[depth] = '0';

Preorder(node->Lnode, depth+1);

key[depth] = '1';

Preorder(node->Rnode, depth+1);

**结果展示：**

**5、心得体会**

这次的大作业主要有一下心得体会：

1、加强了我对于将实际问题转化为一个数学模型的能力，在这次的作业中的问题用例是一个实际的问题：通过给定的权值，为数据设计前缀码，使得代价最小。然后我将这个问题巧妙的转化成了通过给定权值生成霍夫曼树的问题，加强了我对于将实际问题转化为一个数学模型的能力。

2、提高了我解决实际问题的能力，将一个实际问题，抽象化为一个逻辑表达式后，可以借助计算机优秀的计算能力，设计出一个合理的算法，通过计算机解决实际问题，让我感受到了计算机技术的强大。

3、巩固了在课堂上学到的知识，相当于再次复习了书本中逻辑命题中的知识点，比如如何将构造一个霍夫曼树，同时生成霍夫曼编码。同时也拓展了我的知识，通过这次作业我了解到了树的中序遍历的代码实现。

4、提高了我的代码能力。将一个算法实际用代码写出来还是十分考验我的代码能力的，比如实际写代码的过程中我就出现了数组越界等诸多问题，整个作业完成之后，我感觉到我的代码能力得到了提升。