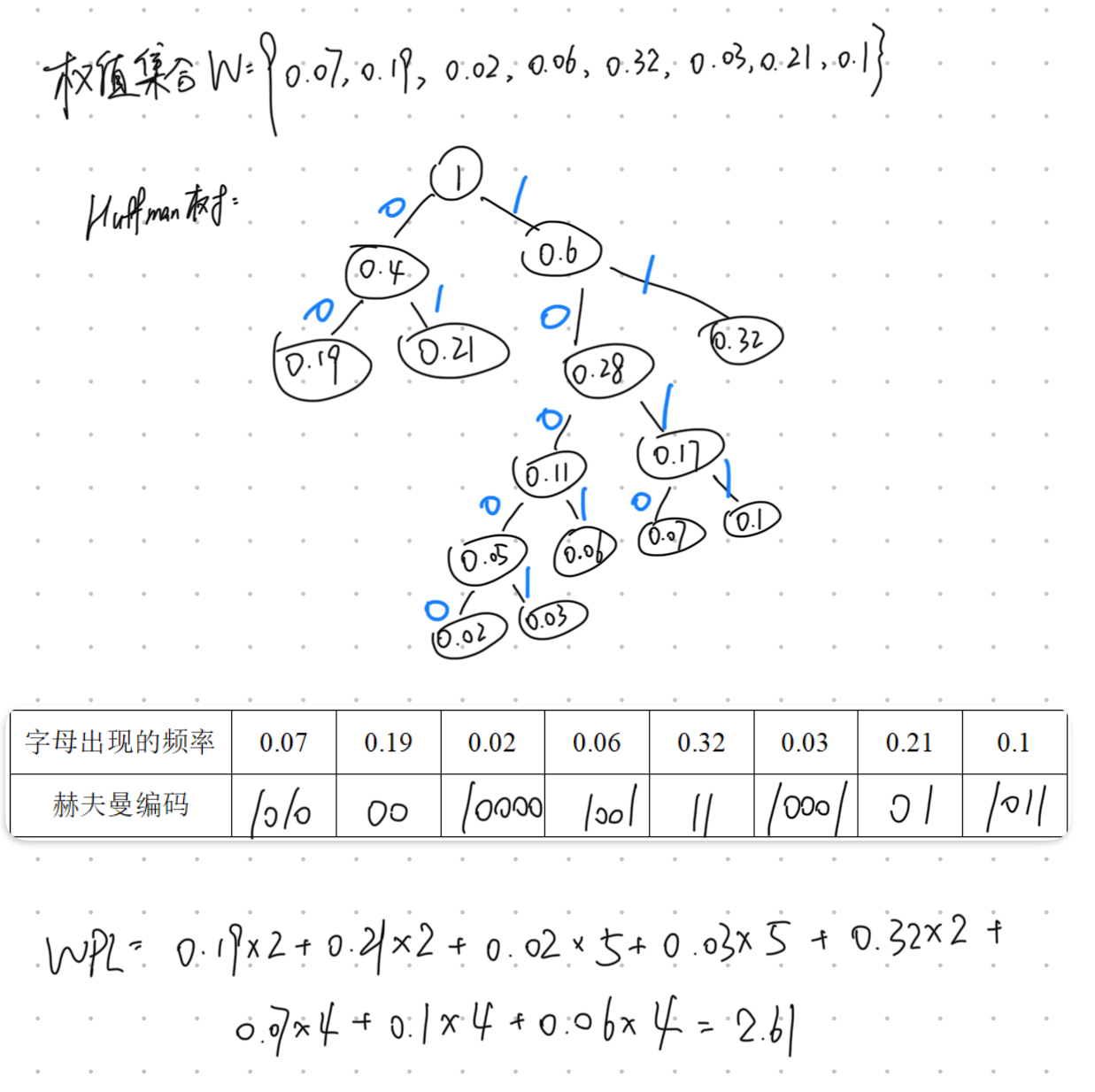
08022311陈鲲龙

1.假定用于通信的电文由8个字母组成，字母在电文中出现的频率分别为0.07，0.19，0.02，0.06，0.32，0.03，0.21，0.1，试为这8个字母设计赫夫曼编码。

* 1. 画出所构造的赫夫曼树（构造过程可略，左分支为0，右分支为1）；
  2. 写出8个字母的赫夫曼编码并计算其带权路径长度；
  3. 关于该问题的另一种编码方案是使用0~7的二进制表示形式，请比较两种方案的优缺点。



二进制编码和赫夫曼编码优缺点比较:

二进制编码用三位二进制数000、001、010、011、100、101、110、111，即可对应8个字母，那么二进制编码的WPL可算出为3\*（0.07+0.19+……）=3\*1（总概率和为1）=3，所以二进制编码的WPL=3大于赫夫曼编码的WPL=2.61，体现了赫夫曼编码对于权值的优化，提升了结点访问效率；但赫夫曼编码也有缺点，就是其编码位数是不同的，也不可以随意用0补齐，否则破坏其路径代表意义，反而二进制编码是非常整齐的三位二进制数，所以在使用时二进制编码可能会比赫夫曼编码更方便，但对于数据量较大，权值差别大的情况，我认为还是要用赫夫曼编码来优化WPL，使其值尽可能的小。

2.算法实现：从二叉树中找出所有结点的最大值并返回，若为空树则返回0。

（提示：基于某种遍历算法。）

算法实现源代码如下：

#include<iostream>

#include<stdio.h>

#include<malloc.h>

#include<string>

using namespace std;

#define ElemType int

int maxnum = 0;//储存树中最大值

class Node//每个结点包含结点存放的数据以及指向此结点的左、右孩子结点的指针

{

public:

ElemType data;

Node\* lchild, \* rchild;

};

void createTree(Node\* &root) {//这里的&十分重要 不可删去 否则值传递无法修改主函数中的tr变量

int ch;

cout << "请输入结点中存放的数据：";

cin >> ch;

if (ch == 0)

{

root = nullptr;

}

else

{

root = new(Node);

root->data = ch;

createTree(root->lchild);//递归，直到结点的左右孩子都被赋0即nullptr才停止

createTree(root->rchild);

}

}

void findmax(int t) {//比大小找树中最大值

if (t > maxnum)

maxnum = t;

}

void preOrder(Node\* root) {//先序

if (root == nullptr) {}

else

{

findmax(root->data);//(三种)遍历时顺便比较找最大值

cout << root->data << " ";

preOrder(root->lchild);

preOrder(root->rchild);

}

}

void inOrder(Node\* root) {//中序

if (root == nullptr) {}

else

{

findmax(root->data);

inOrder(root->lchild);

cout << root->data << " ";

inOrder(root->rchild);

}

}

void postOrder(Node\* root) {//后序

if (root == nullptr) {}

else

{

findmax(root->data);

postOrder(root->lchild);

postOrder(root->rchild);

cout << root->data << " ";

}

}

int main() {

Node\* tr = nullptr;

cout << "先序创建树（需要用0补成完全二叉树！）：\n";

createTree(tr);

cout << "先序遍历：\n";

preOrder(tr);

cout << "\n中序遍历：\n";

inOrder(tr);

cout << "\n后序遍历：\n";

postOrder(tr);

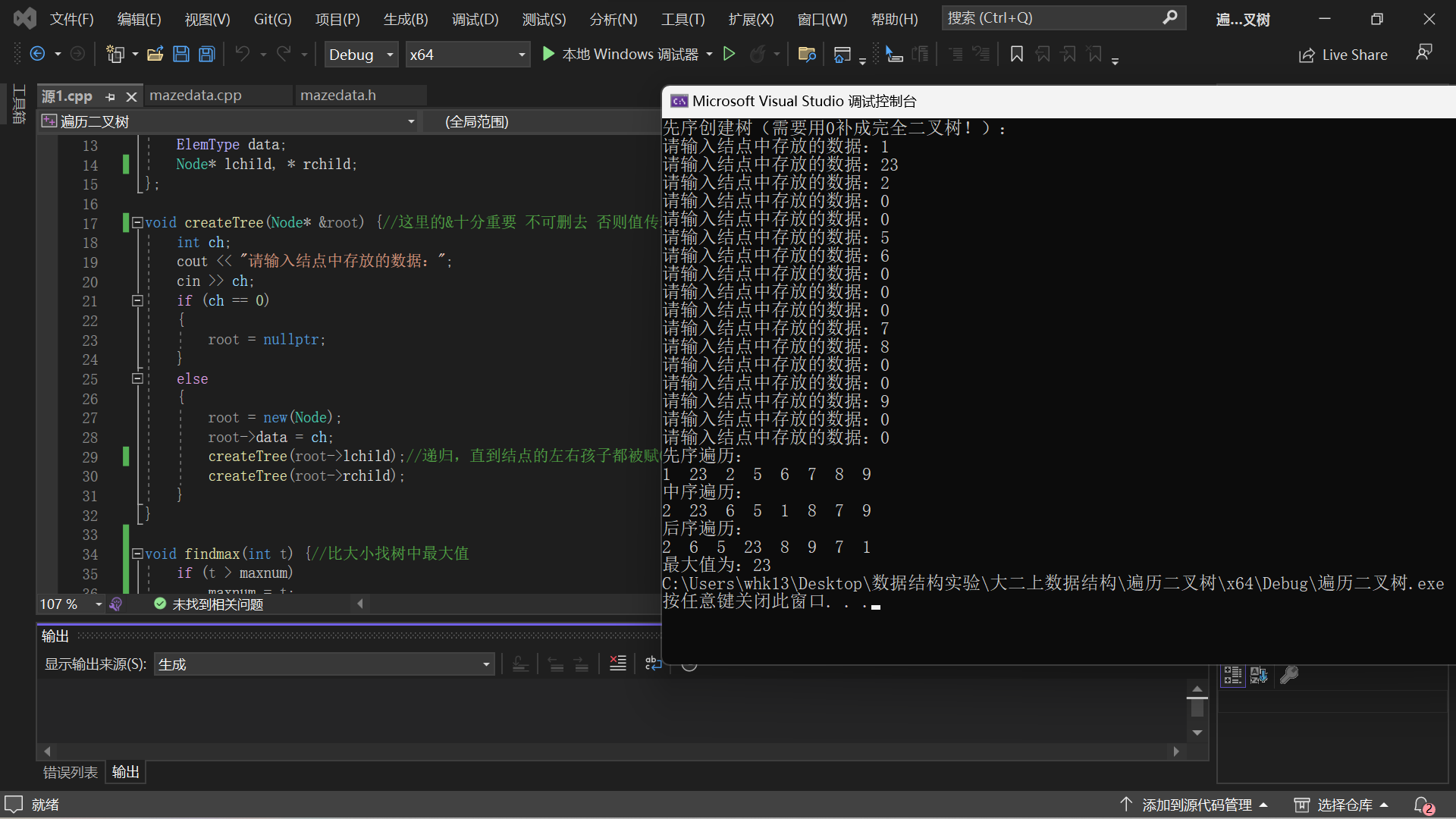
cout << endl;

cout << "最大值为：" << maxnum;

return 0;

}

以一例来阐述二叉树的创建和遍历：



输入逻辑如下图：（输入0仅代表用空指针补成完全二叉树）

