计算机与信息工程学院实验报告

••••••••••••••••••••••••••••••••• 密 ••••••••••••••••••••••••••••••••• 封 ••••••••••••••••••••••••••••••••• 线 •••••••••••••••••••••••••••••••••

姓名：\_扶靖棋\_\_ 学号：\_2012020028\_ 专业：\_计算机科学与技术\_ 年级：\_2020级\_

课程： 数据结构 主讲教师：\_\_杨伟\_\_ 辅导教师：\_杨伟

实验时间：\_2022\_年 \_4\_月 18\_日 \_上\_午\_8\_时至\_10\_时，实验地点\_计算机大楼\_\_

实验题目： 实验四 二叉树的操作（4学时）

实验目的：（1）掌握二叉树的二叉链表存储方式及二叉树的特征；

（2）验证二叉树在二叉链表存储结构下遍历操作的实现；

（3）通过该实验，使学生理解哈夫曼树的概念，掌握哈夫曼树及哈夫曼编码的构造过程，体会网络发送端和接收端编码和译码过程及其工作原理。

实验环境（硬件和软件） Visual Studio Code/Win 10

实验内容： 一、（1）采用二叉链表结构建立二叉树；

（2）编程实现二叉树的先序、中序、后序和层序遍历；

(3） 编程实现非递归中序遍历

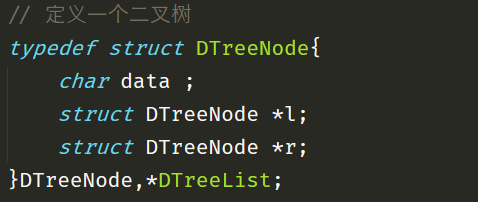
（4）编程实现：求二叉树的高度和叶子结点个数 ；

二、给定报文中26个字母a-z及空格的出现频率{64, 13, 22, 32, 103, 21, 15, 47, 57, 1, 5, 32, 20, 57, 63, 15, 1, 48, 51, 80, 23, 8, 18, 1, 16, 1, 168}，构建哈夫曼树并为这27个字符编制哈夫曼编码，并输出。模拟发送端，从键盘输入字符串，以%为结束标记，在屏幕上输出输入串的编码；模拟接收端，从键盘上输入0-1哈夫曼编码串，翻译出对应的原文。

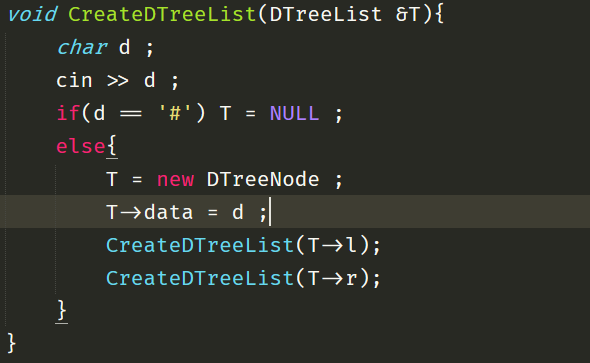
实验步骤：

二叉树的基本操作：

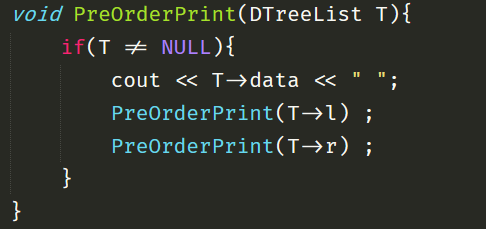
1. 定义一个链表存储的二叉树



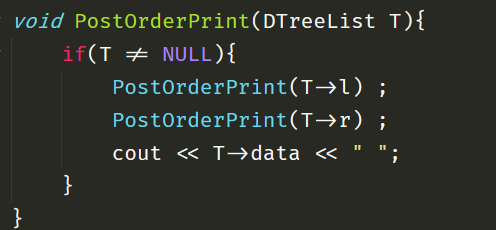
1. 利用前序遍历创建一个二叉树



1. 先序遍历：
   1. 判断当前结点是否为空结点
   2. 先输出当前节点的值
   3. 递归当前结点的左节点和右节点



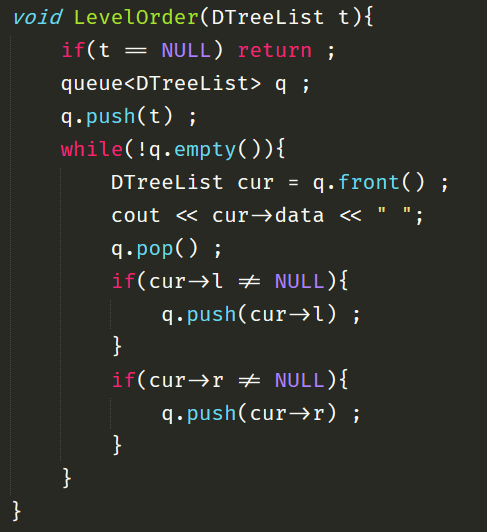
1. 后序遍历：
   1. 判断当前结点是否为空结点
   2. 递归当前节点的左节点
   3. 递归当前结点的右节点
   4. 输出当前结点的值



1. 非递归求中序遍历
   1. 申请一个新栈，记为s，申请一个变量bt，初始时令bt为头节点
   2. 先把bt节点压入栈中，对以bt节点为头的整棵子树来说，依次把整棵树的左子树压入栈中，即不断令bt=bt->left，然后重复步骤b
   3. 不断重复步骤b，直到发现bt为空，此时从s中弹出一个节点记为bt，打印bt->data的值，并让bt = bt->right，然后继续重复步骤b
   4. 当s为空并且bt为空时结束

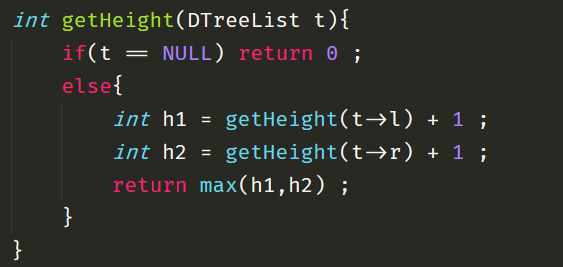


1. 层序遍历
   1. 首先申请一个新的队列，记为q；
   2. 将头结点t压入q中；
   3. 每次从q中出队，记为cur，然后打印cur->data值，如果cur左孩子不为空，则将左孩子入队；如果cur的右孩子不为空，则将右孩子入队；
   4. 重复步骤3，直到q为空。



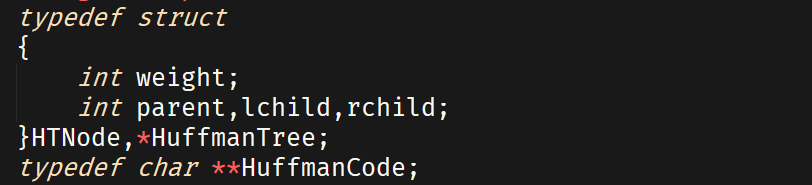
1. 求树高

先判断当前结点是否为空结点，若为空结点，则返回0，代表此节点不存在，否则递归令该根节点的左结点的高度+1记为h1，递归令该根节点的右结点的高度+1记为h2，返回max(h1,h2)得到树的高度。



哈夫曼编码：

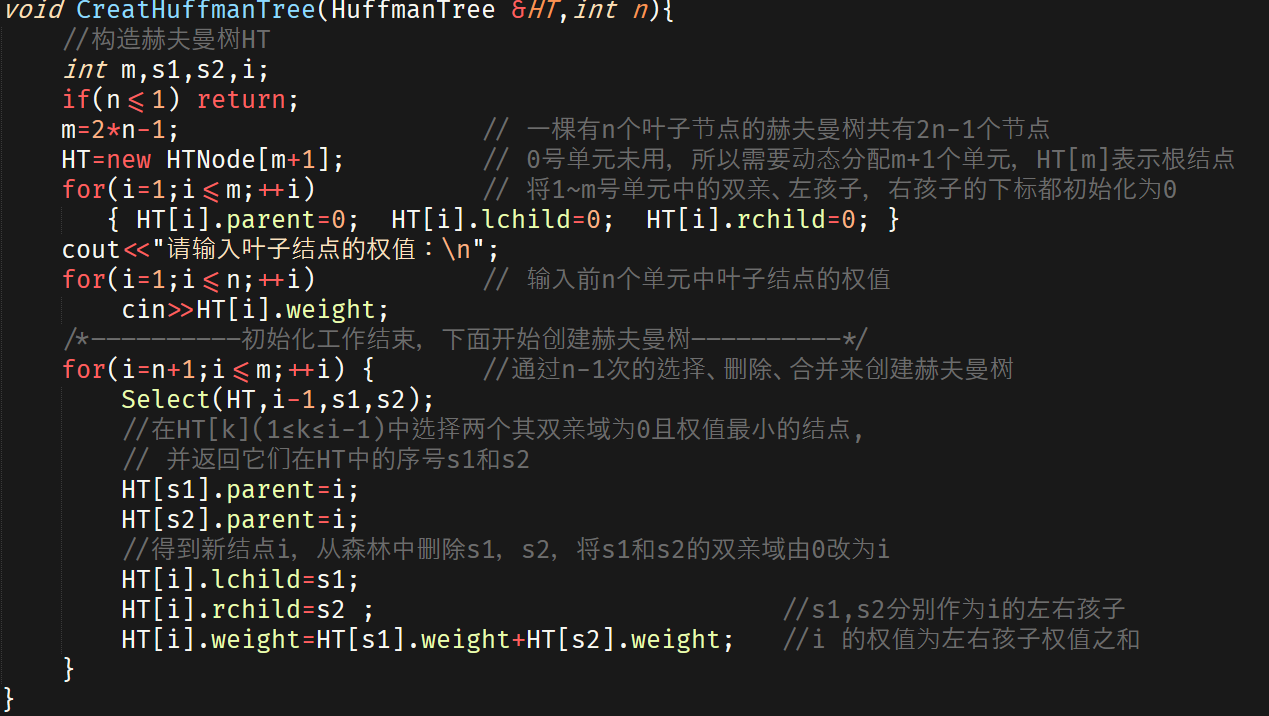
1. 定义哈夫曼树



1. 选择函数的编写
   1. 通过循环遍历找到权重最小的结点与权重第二小的父节点为0的结点，用s1与s2记录这两个未被使用的结点的下标。



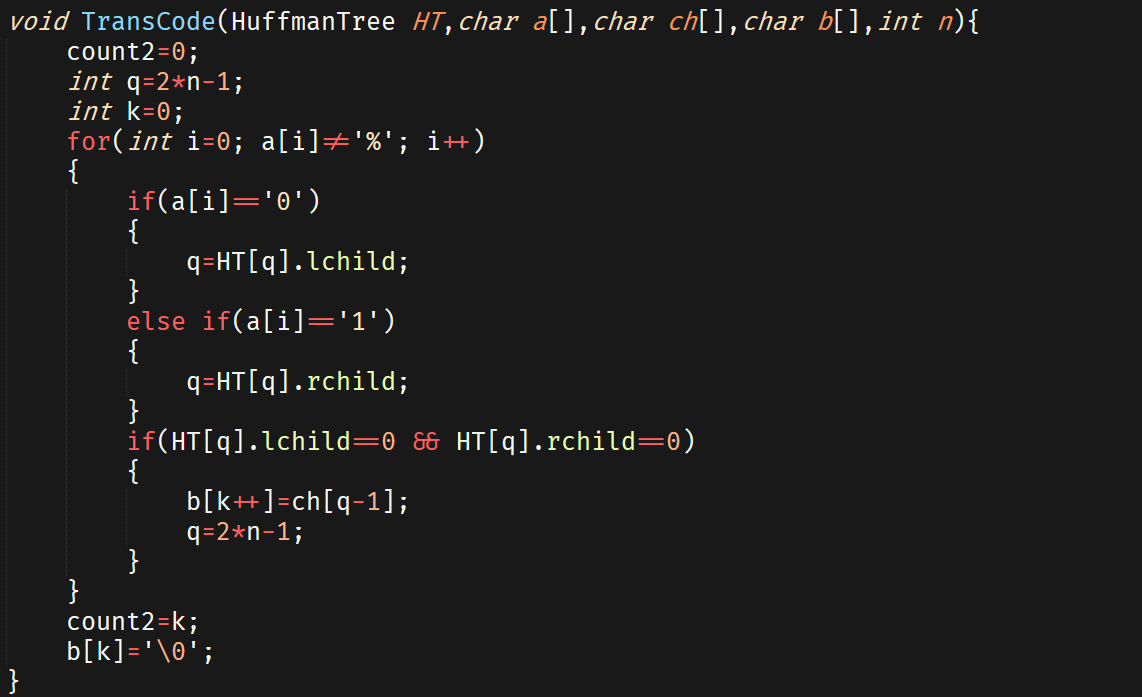
1. 创建哈夫曼树
   1. 初始化：动态申请2\*n个单元；然后循环2n-1次，从1号单元开始，依次将1至2n-1所有单元的双亲，左孩子，右孩子的下标都初始化为0;最后再循环n次，输入前n个单元中叶子结点的权值。
   2. 创建树：循环n-1次，通过n-1次循环，删除与合并来创建哈夫曼树。选择是从当前森林中选择双亲为0且权值最小的两个数根结点s1和s2；删除是指将结点s1和s2的双亲改为非0；合并就是将s1和s2的权值和作为一个新节点的权值依次存入到数组的第n+1之后的单元中，同时记录这个新结点左孩子的下标为s1，右孩子下标为s2。



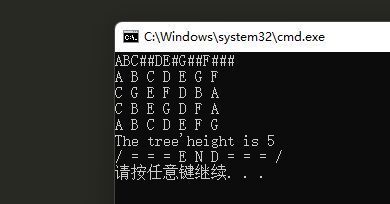
1. 创建哈夫曼编码
   1. 从叶子到根逆向求每个字符的赫夫曼编码，存储在编码表HC中

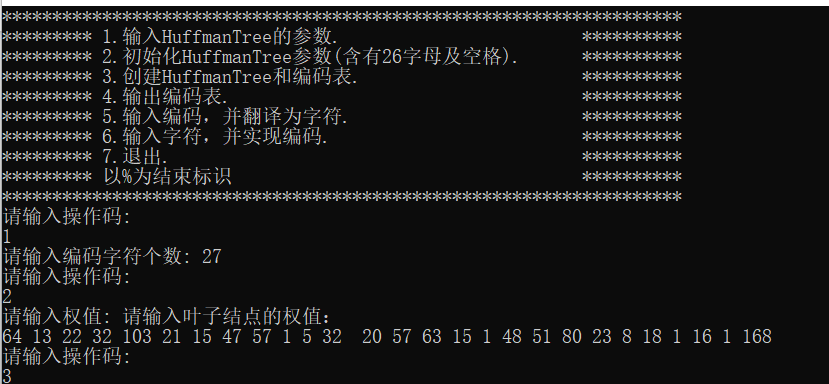


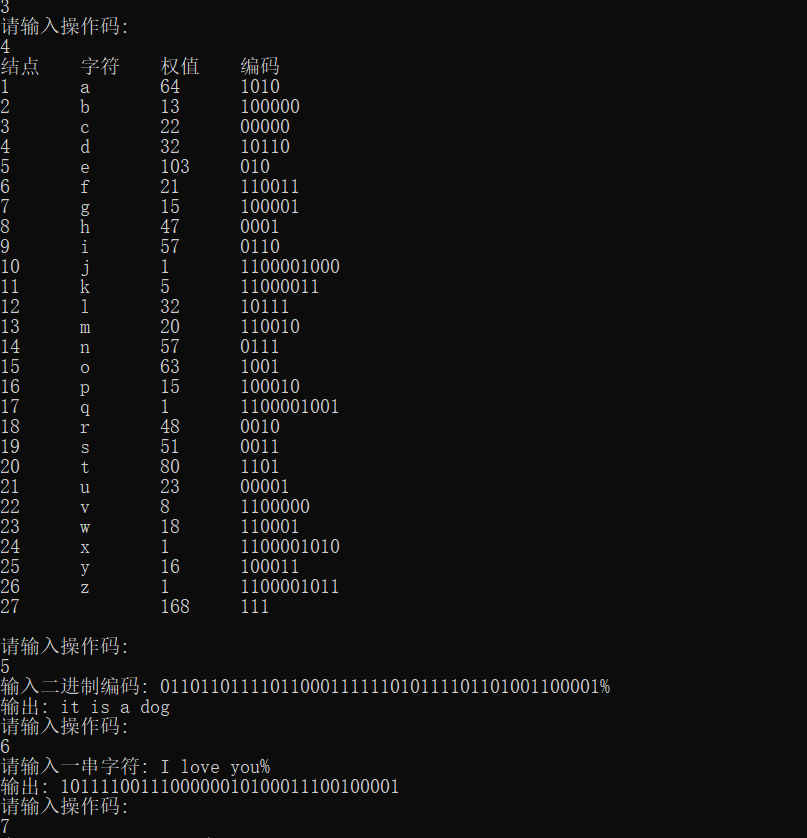
5．哈夫曼编码转码



实验数据记录：







问题讨论：

问题一：在编写层序遍历二叉树的算法时，输出结果进入死循环，重复输出某个字母，不能得到预期结果。

分析原因：经过不断debug，发现是因为在输出队列q的头元素后没有将它pop出去，导致q的头元素无法出队，使得队列始终存在元素，不能更新。

问题二：在编写select函数时，寻找最权值第二小的结点时，导致最终s1，s2存储的下标相同。

分析原因：在寻找到最小权值的节点后，由于寻找的是权值第二小的结点，原有节点未被更改父节点的值，因此在寻找权值第二大的结点时，寻找到的结点仍是第一小的结点。

问题三：构造哈夫曼树后，进行哈夫曼编码，得到的二进制编码发生错误。

原因分析：混淆了结点下标与节点权值，未将找到的俩节点的权值相加赋值给其父节点。