数据结构与算法基础课程实验报告

实验 2：二叉树及其应用

姓名	李大鑫	院系	计算机学院	学号	1170300825
任课教师	臧天仪		指导教师	臧天仪	
实验地点	格物 207		实验时间	11/24/2018	
实验课表现	出勤、表现得分 10%		实验报告得分 40%		实验总分	
	操作结果得分 50%					

实验目的：

1. 掌握树的链式存储方式及其操作实现（创建、遍历、查找等）。
2. 掌握二叉树用不同方法表示所对应的不同输入形式。
3. 掌握二叉树中各种重要性质在解决实际问题中的应用。
4. 掌握哈夫曼树的构造方法及其编码方法。
5. 掌握二叉排序树的特性及其构造方法。

实验内容：

假设自上而下按层次，自左至右输入每个结点的一个三元组(N, P, L/R)。其中 N 为本结点的元素，P 为其父结点，L 指示 N 为 P 的左孩子，R 指示 N 为 P 的右孩子。试写一个建立二元树在内存的双链表示算法，并实现先根、中根、后根以及层序遍历算法。

实验要求：（学生对预习要求的回答）（5 分）　　得分：

1、 编写建立二叉树的二叉链表存储结构（左右链表示）的程序，并以适当的形式显示和保存二叉树？

使用链表的形式实现整个二叉树结构的存储，通过左右指针链接左右子节点，节点中保存的信息有字符数组data代表整个树节点的值，depth代表当前节点在树中的深度，fa指针链接父节点。树的显示通过三种遍历间接表示。

2、 采用二叉树的二叉链表存储结构，编写程序实现二叉树的先序、中序和后序遍历的递归和非递归算法以及层序遍历算法，并以适当的形式显示和保存二叉树及其相应的遍历序列？

三种顺序的遍历问题的递归实现可以通过函数递归然后调整输出与递归的顺序简单实现。对于三种遍历的非递归实现：以前序遍历为例，首先使用一个栈来存储访问到的节点，首先将根节点入栈，[“以上相同操作”开始]取出栈顶指针，当节点的左子树不为空时，不停的向下进入其左子树，将节点入栈，直到左子树为空，此时取出栈顶节点，对右节点进行以上相同操作。以上原理可以简单理解为**每棵子树都可以分解为长度不一的一串左儿子的遍历**。对于后序遍历，改变为不停向下进入其右子树。对于中序遍历，先不停地向下访问儿子，左子树没有就访问右子树，当访问到左右子树都空的时候就开始输出返回，将栈中的节点弹出，对弹出节点进行相同的操作。

3．给定一个二叉树， 编写算法完成下列应用:（二选一）

（1）判断其是否为完全二叉树；

（2）求二叉树中任意两个结点的公共祖先。

1）对树进行 bfs 访问，用一个数组记录访问到的节点性质，当左右子树都为空的时候向数组中添加 00，如果有不为空的则添加 1，这样当 bfs 结束之后检查产生的数字序列，只有当前面全是 1 后面全是 0 的序列才代表这棵树是完全二叉树，其他情况均代表这棵树不是完全二叉树。

2）首先对这棵树进行深度优先搜索，搜索过程中产生每个节点的 depth,fa 信息，利用这两个信息寻找最近公共祖先。首先将两个节点调整到同一高度（利用 fa、depth），如果相同则为最近公共祖先，否则，将两个节点同时向上遍历直到相同，则该节点就是最近公共祖先。

实验过程中遇到的问题如何解决的？（10 分）（着重从软件调试、质量保证、结果分析方面进行阐述）	得分：

问题 1：

实验中的主要问题集中在实现非递归方式的三种遍历，其中的细节实现容易出错。我使用的软件调试方法是直接打印访问序列，通过与自己建树应该出现的访问顺序进行比较，首先保证访问顺序的正确性。当不确定实际应该打印的位置的时候，我进行了多种尝试。具体的实现原理如下：以前序遍历为例，首先使用一个栈来存储访问到的节点，首先将根节点入栈，[“以上相同操作”开始]取出栈顶指针，当节点的左子树不为空时，不停的向下进入其左子树，将节点入栈，直到左子树为空，此时取出栈顶节点，对右节点进行以上相同操作。以上原理可以简单理解为每棵子树都可以分解为长度不一的一串左儿子的遍历。对于后序遍历，改变为不停向下进入其右子树。对于中序遍历，先不停地向下访问儿子，左子树没有就访问右子树，当访问到左右子树都空的时候就开始输出返回，将栈中的节点弹出，对弹出节点进行相同的操作。

问题 2：

对于检查是否是完全二叉树的方法，简单来看的话就是只允许倒数第二层和倒数低一层的右侧缺一块。这样的性质，正好可以使用 bfs 访问节点，然后检查子节点的方式来验证。实现如下：对树进行 bfs 访问，用一个数组记录访问到的节点性质，当左右子树都为空的时候向数组中添加 00，如果有不为空的则添加 1，这样当 bfs 结束之后检查产生的数字序列，只有当前面全是 1 后面全是 0 的序列才代表这棵树是完全二叉树，其他情况均代表这棵树不是完全二叉树。

问题 3：

如何读文件也是需要解决的问题：因为读入的节点的值必须要兼容字符串所以需要为节点在 struct 中声明一个字符串域来承载。

本次实验的体会（结论）（10 分）	得分：

本次实验的主要目的就是实现一棵二叉树、前序中序后序遍历的递归和非递归实现、判断是否是完全二叉树、寻找两个节点的最近公共祖先。 现在将本次实验的体会总结如下：

1） 二叉树的三种遍历方式的非递归实现的方法：这种非递归的实现方法非常值得学习，主要就是使用一个栈来完成“回溯”的功能，其中这种将树分成许多个长短不一的左孩子串的思想非常值得借鉴，感觉与“分治”的思想类似

2） 当有了实现思路的时候，如何更简单优雅的实现也是一门艺术：比如，当检查数字串是否是前面全 1 后面全 0 的时候完全可以只是用一个 flag 标志位来实现；还比如，当寻找最近公共祖先的时候，假设 a 是较高的节点，如果不满足，我们交换 ab 使得满足条件，这样就减少了算法执行时候的讨论。

3） 一个实验程序必须要满足基本的交互。

思考题：（15 分）

思考题 1：（5 分）	得分：

思考题 1：在二元树的表示中通常都有哪几种方法？其特点是什么？

1、 二叉树的左右链表示

在每个节点中设置两个域 lchild 和 rchild，使他们分别指向该节点的左子树和右子树，若没有这设置为 NULL。

特点：表示简单，只需要使用一个根节点的指针就可以获得整个树。

2、 二叉树的游标表示

类似于左右链表示法，所不同的是指向每个节点的指针用游标来表示，需要使用一个结构数组来存储所有的节点，游标就是数组的下标。

特点：需要额外使用一个数组来表示一棵树。链接左右孩子的时候比较方便。

3、 线索二叉树

与二叉树的左右链表示类似，当某个节点的 lchild 为空的时候，则令其指向该节点的中序遍历的前导节点；当某个节点的 rchild 为空的时候，则令其指向该节点的中序遍历的后继节点。为此增设两个标志位 ltag 和 rtag 来表示两个孩子域是否是真的指向孩子节点。

特点：线索二叉树的优点是使对二叉树的遍历操作比较简单。但是写代码的时候需要特别判断一下。

4、完全二叉树的数组表示法

从根开始，自左向右，自上而下进行标号，然后把节点按照标号的顺序将其依次存放一位数组 TREE 中。

特点：这种表示法对于完全二叉树中的任意一个节点 i，可以很容易地求出其父亲、左儿子和右儿子的位置。

思考题 2：（5 分）	得分：

思考题 2：在树的表示中通常都有哪几种方法？其特点是什么？

1、 双亲表示法：双亲表示法通过记录自己的父亲来实现节点之间的连接关系，比如并查集就是用这种方法进行连接的。特点：简单方便，只需要记录父节点存储空间少；需要记录所有的节点，否则就可能丢失这个节点（访问不到）；适用于查找祖先的需求。

2、 孩子表示法：在树中，每个结点有若干个指针域，指向其孩子。由于孩子的数目不一，所以可以有两种表示方法：

方法一： data child1 child2 …… childd

特点：这种表示法，所有的结点都是同构的，其中 d 为树的度。由于树中很多结点的度小于 d，所以浪费了很多空间

方法二： data degree child1 child2 …… childn

特点：在这种表示法中，degree 指示了该结点有几个孩子，n 为结点的度，不同的结点所占的格式不同，所以能够节约空间，但操作不便。

树的另一种常用的表示方法就是**儿子链表**表示法。这种表示法用一个线性表来存储树的所有结点信息，称为结点表。对每个结点建立一个儿子表。儿子表中只存储儿子结点的地址信息，可以是指针，数组下标甚至内存地址。由于每个结点的儿子数目不定，因此儿子表常用单链表来实现。

3、 孩子兄弟表示法：

树的左儿子右兄弟表示法又称为二叉树表示法或二叉链表表示法。每个结点除了 data 域外，还含有两个域，分别指向该结点的最左儿子和右邻兄弟。这种表示法常用二叉链表实现，因此又称为二叉链表表示法。

特点：用树的左儿子右兄弟表示法可以直接实现树的大部分操作，只有在对树结点作 Parent 操作时需遍历树。如果要反复执行 Parent 操作，可在结点记录中再开辟一个指向父结点的指针域，也可以利用最右儿子单元中的 Right_Sibling 作为指向父结点的指针(否则这里总是空指针)。当执行 Parent(v)时，可以先通过 Right_Sibling 逐步找出结点 v 的最右兄弟，再通过最右兄弟的 Right_Sibling(父亲指针)找到父结点。这个结点就是结点 v 的父亲。在这样的表示法下，求一个结点的父亲所需要的时间正比于该结点右边的兄弟个数。不过，这时每个记录中需要多用一位(bit)空间，用以标明该记录中的 right_sibling 是指向右邻兄弟还是指向父亲。

思考题 2：（5 分）	得分：

思考题 3：我们讨论森林和二元树之间的转换，其目的是什么？

将森林转换成为二元树之后，对于二叉树所提出的算法，完全可以被森林所用而无需进行修改。

从森林和二叉树之间的对应关系可以看到，对森林遍历的先序遍历和后序遍历的定义正好与这主公自然的对应关系相符合。这是因为第一棵树的所有子树对应于二叉树的左子树，而其余的树则对应于二叉树的右子树，将这些定义和关于二叉树节点访问顺序的相应定义相比较可以看出，将先序遍历对一个森林进行遍历操作，与按照先序遍历对对应的二叉树进行遍历的操作完全一致，因此对于二叉树所提出的算法，完全可以被森林所用而无需进行修改。

指导教师特殊评语：

指导教师签字：
日期：