**实 验 报 告 （ 2 ）**

|  |  |
| --- | --- |
| **实验名称**：树的应用 | **实验地点**： |
| **所使用的工具软件及环境：** | |
| **一、实验目的：**  1、掌握二叉树的结构特征，以及各种存储结构的特点及使用范围。  2、掌握用指针类型描述、访问和处理二叉树的运算。  3、掌握树的应用算法。 | |
| **二、实验内容及要求：**（填写题目内容及输入输出要求）  **1. 编写程序，根据给定的一棵二叉树的后序遍历和中序遍历结果：**  **（1）输出该二叉树的前序遍历结果；**  **（2）判断给定的二叉树是否是二叉搜索树。**  1）设计思路  1.方法一：通过模拟实现，两序列上下对照。先后确定根节点，左子树和右子树。时间复杂度分析为O（N\*N）。（线性表查找的时间复杂度为O（N），共需要处理N个节点）。  2.方法二：通过中序后序表达式补位（标记叶子节点）后进行递归创建（或者层序创建），该方法最为简便且常用。（创建过程时间复杂度O（N），补位过程时间复杂度O（N））  对于其是否为二叉搜索树的判断，我们只需要判断其中序遍历序列是否为不下降序列即可，该操作时间复杂度为O（N）  2）数据结构的描述  Tree是n个节点构成的有限集合。N=0时，称之为空树；n非零时，其余节点可以化为m个不相交的子集，子集称之为子树的根节点。每个子树根节点有且仅有一条边与其父亲节点相连。  3）程序代码    **2.编写算法计算二叉树最大的宽度（二叉树的最大宽度是指二叉树所有层中结点个数的最大值）。**  1）设计思路  1.方法一：通过迭代的方法将先序序列（含叶子节点标记）转换到一维数组存储。而后对每一层进行层序遍历，统计宽度即可。时间复杂度为O（N）。  2.方法二：通过递归的方法从头到尾遍历，中间使用level标记层数，进行统计，时间复杂度为O（N）。  （下采用第一种方法进行）  2）数据结构的描述  Tree是n个节点构成的有限集合。N=0时，称之为空树；n非零时，其余节点可以化为m个不相交的子集，子集称之为子树的根节点。每个子树根节点有且仅有一条边与其父亲节点相连。  3）程序代码  **3.设顺序存储的二叉树中有编号为i和j的两个结点，请设计算法求出它们最近的公共祖先结点的编号。**  1）设计思路  顺序存储时候，下标运算有天然优势。可以通过下标运算找到最近的父亲节点。  2）数据结构的描述  无需数据结构，简单判断即可；用二进制位运算可以大幅降低时间复杂度和代码量  3）程序代码 | |
| **三、程序运行结果（分析计算时空复杂度，运行界面截图）**     1. 涉及到递归遍历，每个节点创建时候仅访问一次，时间复杂度为O（N）；   **（4 3 1 2）为存储后先序遍历的结果。1 表示是一颗二叉搜索树**  **2.向线性表填充过程，层际比较复杂度均为O（N）。整体复杂度为O（N）**    **3.二进制移位过程中复杂度为O(LogN)，只有这一步耗时操作。整体复杂度O（LogN）** | |

**成绩：\_\_\_\_\_\_\_\_\_\_\_\_** **任课教师签名：\_\_\_\_\_\_\_\_\_\_\_\_** 2025年 月 日