数据结构第二次作业

班级：计科1194 姓名：万一凡 学号：201911111420

P246 3.为实现以下各种功能，x结点表示该结点的位置，给出树的最适合的存储结构

1. 求x和y结点的最近祖先结点： **双亲存储结构**
2. 求x结点的所有子孙结点： **孩子链存储结构**
3. 求根结点到x结点的路径： **双亲存储结构**
4. 求x结点的所有右边兄弟结点： **孩子兄弟连存储结构**
5. 判断x结点是否为叶子结点:  **孩子链存储结构**
6. 求x结点的所有孩子结点: **孩子链存储结构**

P246 4.设二叉树bt的一种存储结构如表所示，其中，bt为树根结点指针，lchild,rchild为结点的左，右孩子指针域，在这里使用结点编号作为指针域值，0表示指针域为空；data为结点的数据域，请完成下列各题

1. 画出二叉树bt的树形表示：
2. 写出按先序中序后序遍历二叉树bt所得的结点序列：
3. 画出二叉树bt的后序线索树：

**(1)**

**(2)**

**前序：a b c e d f h g i j**

**中序：e c b h f d j i g a**

**后序：e c h f j i g d b a**

**(3)**

P247 13.假设二叉树中的每个结点值为单个字符，采用二叉链存储结构存储。设计一个算法求二叉树种b种最小值的结点值。

P247 17.假设二叉树种的每个结点值为单个字符，采用二叉链存储结构存储。设计一个算法，采用先序遍历方法求二叉树b种值为x的结点的子孙结点，假设值为x的结点时唯一的。

下附运行情况和源代码：

**13.**

**char min(tree \*p,char &pp){//主函数部分pp可以初始化为根结点**

**if(p!=NULL){//树不为空时执行**

**if(p->data<pp){//对根结点的比较**

**pp=p->data;**

**}**

**if(p->lchlid!=NULL&&pp>p->lchild->data){//如果左孩子存在则比较替换**

**pp=p->lchild->data;**

**}else if(p->rchild!=NULL&&pp>p->rchild->data){//如果右孩子存在比较替换**

**pp=p->rchild->data;**

**}**

**pp=min(p->lchild,pp); // 遍历左孩子处理**

**pp=min(p->rchild,pp); // 遍历右孩子处理**

**}**

**return pp;//返回最小的字节值**

**}**

**17.void xianxu(tree \*p){//先序遍历**

**if(p!=NULL){**

**printf("%c ",p->data);**

**xianxu(p->left);**

**xianxu(p->right);**

**}**

**}**

**void findson(tree \*p,char pp){**

**if(p!=NULL){//树不为空执行**

**if (p->data==pp){//判断结点值 按照先序定义根-左-右**

**xianxu(p->left); //想等先序遍历左子树**

**xianxu(p->right); //先序遍历右子树**

**}**

**findson(p->left, pp);//递归在左子树找**

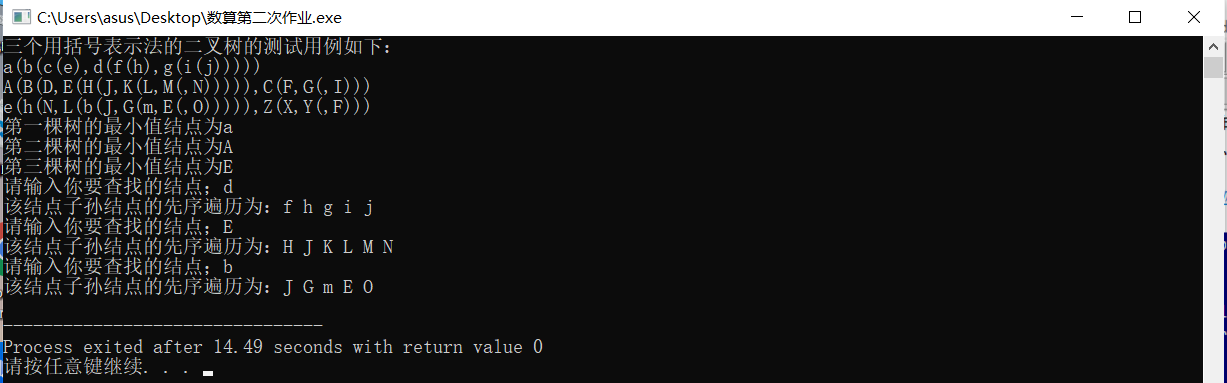
**findson(p->right, pp);//递归在右子树找**

**}**

**}**

**主函数部分省略；**

**运行图如下**



**其中三例测试的二叉树图简单如下**