第三周

4.4树、森林和二叉树的转换

1. 树转化成二叉树，方法如下：
2. 树中所有相邻兄弟之间加一条连线
3. 对树中的每个结点，只保留它与第一个孩子结点之间的连线，删去它与其他孩子结点之间的连线
4. 以树的根结点为轴心，将整棵树顺时针转动一定的角度，使之结构层次分明

树作这样的转换所构成的二叉树是唯一的

一棵树采用孩子兄弟表示法所建立的存储结构与它所对应的二叉树的二叉链表存储结构是完全相同的。

根据树与二叉树的转换关系以及树与二叉树遍历的操作定义：

1. 树的先序遍历=二叉树的先序遍历
2. 树的后序遍历=二叉树的中序遍历
3. 森林转化为二叉树，方法如下：
4. 将森林中的每棵树转换成相应的二叉树
5. 第一课二叉树不动，从第二棵二叉树开始，依次把后一棵二叉树的根结点作为前一棵二叉树根结点的右孩子，当所有二叉树连起来后，此时所得到的二叉树就是由森林转换得到的二叉树
6. 二叉树转化为树和森林
7. 若某结点是其双亲的左孩子，则把该结点的右孩子、右孩子的右孩子......都与该结点的双亲结点用线连起来
8. 删去原二叉树中所有的双亲结点与右孩子结点的连线
9. 整理由(1)、(2)两步所得到的树或森林，使之结构层次分明

4.6二叉搜索树

1. 二叉树的基本操作

性质：二叉搜索树的任何一个结点，设其值为K，则该结点的左子树的任意一个结点的值都小于K；该结点右子树的任意一个结点的值都大于或等于K。又名二叉排序树。

算法实现：

1. MakeEmpty(T)

{ if(T!=NULL)

{ MakeEmpty(T->Left);

MakeEmpty(T->Right);

free(T);

}

return RULL;

}

1. Find操作返回指向树T中具有关键字X的结点指针，结点不存在返回NULL。
2. 若搜索树为空，查找失败
3. 搜索树非空，将给定值X与查找树的根结点关键吗比较
4. 若相等，查找成功，结束查找过程，否则：
5. 当X小于根结点关键码时，查找将在以左孩子结点为根的子树上继续进行
6. 当X大于根结点关键码时，查找将在以右孩子结点为根的子树上继续进行
7. FindMin和FindMax分别返回树中最小元素和最大元素的位置。FindMin从根开始，并且只要有左孩子就向左进行，终止点即为最小的元素。FindMax除分支除分支朝向右孩子之外其余过程相同。
8. Insert 如果X存在则什么也不做，否则，先定位后插入，并返回新树根指针
9. Delete

先找要删结点，若查找失败，直接返回；若找到要删结点，则有以下几种可能：

1. 叶子：立即删除
2. 有一个孩子：在其双亲结点调整指针绕过该结点后删除
3. 两个孩子：用其右子树的最小的数据元素结点值代替该结点的数据，并递归地删除那个替代结点，即递归地用有一个孩子的方法删除