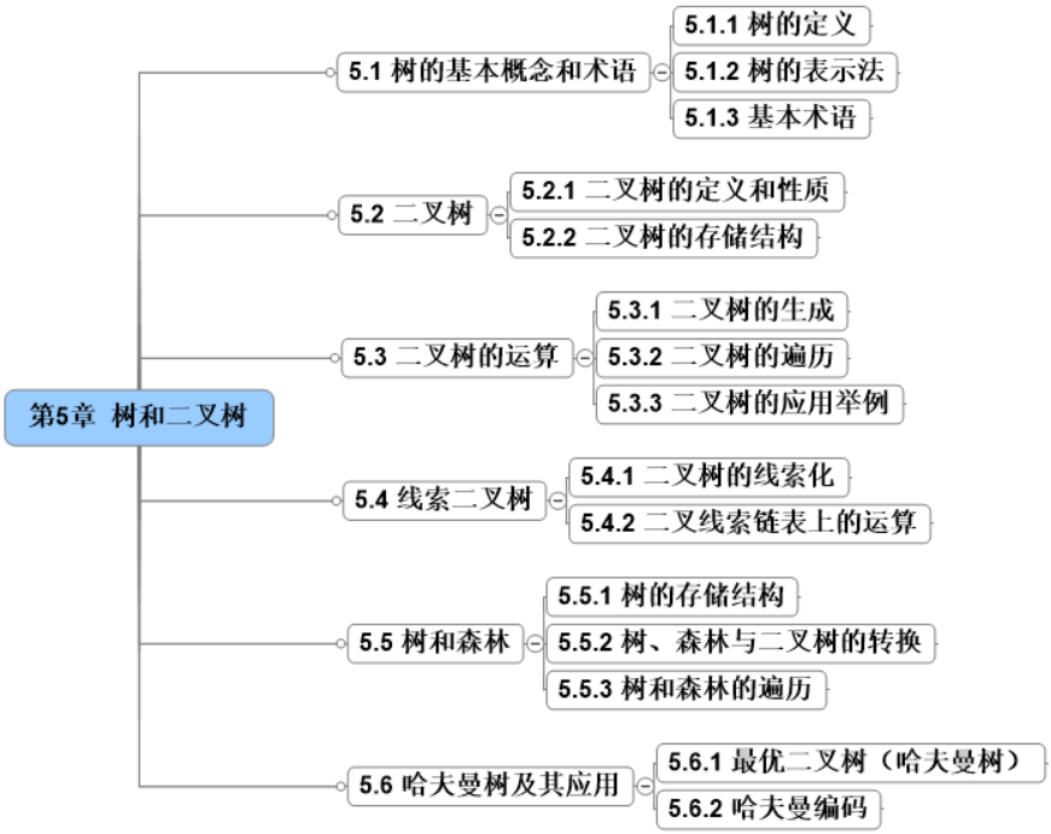
数据结构第七节课官方笔记

目录

一、课件下载及重播方法

二、教材结构图



三、本章知识点及考频总结

（一）选择题（共 9 道）

1. 树（Tree）是 n(n≥0)个结点的有限集 T。它或是空集（空树即 n=0），或者是非空集。

对于任意一棵非空树：

（1）有且保有一个特定的称为根（Root）的结点；

（2）当 n>1 时，其余的结点可分为 m(m>0)个互不相交的有限集 T_1, T_2, ·, T_m，其中每个集合本身又是一棵树，并称为根的子树。

2. 树的结点包含一个数据元素及若干个指向其子树的分支。一个结点拥有的子树数称为该结点的度（Degree）。一棵树中结点的最大度数称为该树的度。度数为零的结点称为叶子结点或终端结点。度数不为零的结点称为非终端结点或分支结点。除根结点之外，分支结点也称为内部结点，而根结点又称为开始结点。

3. 二叉树（Binary Tree）是 n(n≥0)个元素的有限集合，它的每个结点至多只有两棵子树。它或者是空集，或者是由一个根结点及两棵互不相交的分别称作这个根的左子树和右子树的二叉树组成。

4. **性质 1**　在二叉树的第 i 层上至多有 2^{i-1} 个结点（i≥1)。

性质 2　深度为 k（k≥1）的二叉树至多有 2^k-1 个结点。

性质 3　对任何一棵二叉树 T，若其终端结点数为 n_0，度数为 2 的结点数为 n_2，则 $n_0=n_2+1$。

满二叉树：一棵深度为 k 且有 2^k-1 个结点的二叉树称为满二叉树。

完全二叉树：若一棵深度为 k 的二叉树，其前 k-1 层是一棵满二叉树，而最下面一层（即第 k 层）上的结点都集中在该层最左边的若干位置上，则称此二叉树为完全二叉树。

满二叉树一定是完全二叉树，但完全二叉树则不一定是满二叉树。

性质 4　具有 n 个结点的完全二叉树的深度为⌊log n⌋ +1

5. 对于完全二叉树，假设编号为 i 的结点 q_i (0≤i<n)，那么，

①若 i=0，则 q_i 为根结点，无双亲；否则 q_i 的双亲结点编号为 ⌊(i-1)/2⌋。

②若 2i+1<n，则 q_i 的左孩子结点编号为 2i+1；否则 q_i 无左孩子，即 q_i 必定是叶子结点。

③若 2i+2<n，则 q_i 的右孩子结点编号为 2i+2；否则，q_i 无右孩子。

6. 二叉树顺序存储结构仅适用于完全二叉树，但对于一般的二叉树来说，不但会浪费存储空间，而且当经常在二叉树中进行插入或删除结点操作时，需要移动大量的结点。因此，在一般情况下，多采用链式存储方式来存储二叉树。

7.

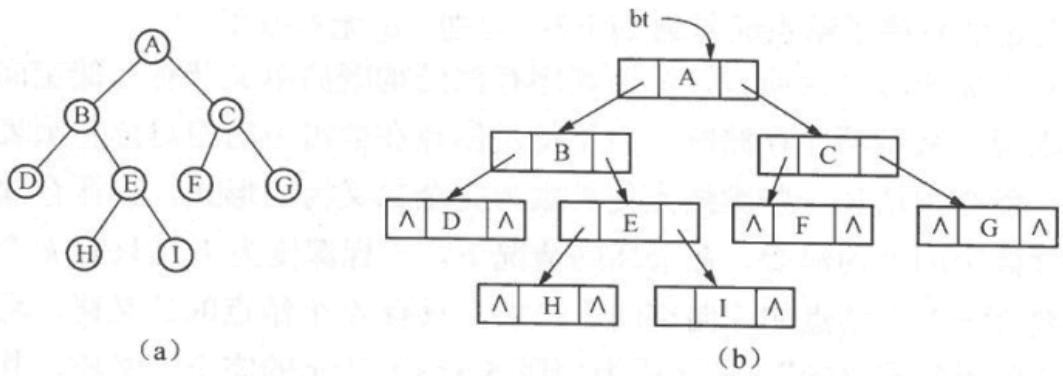


图 5.8　二叉树的链式存储结构示意图

8. **三种遍历的递归算法定义：**

（1）前序遍历二叉树的递归定义

若二叉树非空，则依次进行操作：

①访问根结点；②前序遍历左子树；③前序遍历右子树。

（2）中序遍历二叉树的递归定义

若二叉树非空，则依次进行操作：

①中序遍历左子树；②访问根结点；③中序遍历右子树。

（3）后序遍历二叉树的递归定义

若二叉树非空，则依次进行操作：

①后序遍历左子树；②后序遍历右子树；③访问根结点。

9. 已知一棵二叉树的前序和中序遍历序列或中序和后序遍历序列，可唯一地确定一棵二叉树。

（二）主观题（共 3 道）

1. 用 C 语言描述的前序遍历的递归算法如下：

```
void Preorder ( BinTree bt )
{    //采用二叉链表存储结构，并设结点值为字符型
   if(bt!=NULL) {
        printf("%c", bt->data); //访问根结点
        Preorder ( bt->lchild ); //前序遍历左子树
        Preorder ( bt->rchild );//前序遍历右子树，r 表示返回点
}
```

2.

```
void Inorder (BinTree bt)
{   //中序遍历二叉链表算法
   if(bt!=NULL) {
        Inorder(bt->lchild);          //中序遍历左子树
        printf("%c", bt->data)        //访问根结点
        Inorder(bt->rchild);          //中序遍历右子树
      }
}
```

```
3. void Postorder(BinTree bt)
{   //后序遍历二叉链表算法
    if(bt!=NULL) {
        Postorder(bt->lchild);          //后序遍历左子树
        Postorder(bt->rchild);          //后序遍历右子树
        printf("%c", bt->data);         //访问根结点
    }
 }
```

四、配套练习题

1、已知在一棵度为 3 的树中,度为 2 的结点数为 4,度为 3 的结点数为 3,则该树中的叶子结点数为(　　)。

A: 5

B: 8

C: 11

D: 18

2、在一棵二叉树中，度为 2 的结点数为 15，度为 1 的结点数为 3，则叶子结点数为(　　)。

A: 12

B: 16

C: 18

D: 20

3、若一棵具有 n(n>0)个结点的二叉树的先序序列与后序序列正好相反，则该二

叉树一定是(　　)。

A: 结点均无左孩子的二叉树

B: 结点均无右孩子的二叉树

C: 高度为 n 的二叉树

D: 存在度为 2 的结点的二叉树

[参考答案]：CBC

五、其余课程安排