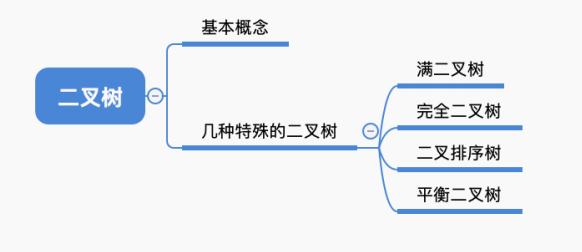
二叉树定义和基本术语



二叉树的基本概念：

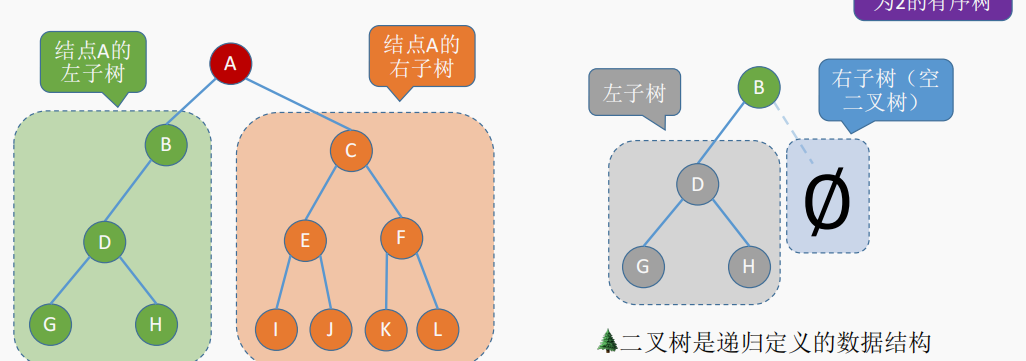
二叉树是n（n≥0）个结点的有限集合：

① 或者为**空二叉树**，即n = 0。

② 或者由**一个根结点**和**两个互不相交**的被称为根的**左子树**和**右子树**组成。左子树和右子树又分别是一棵二叉树。

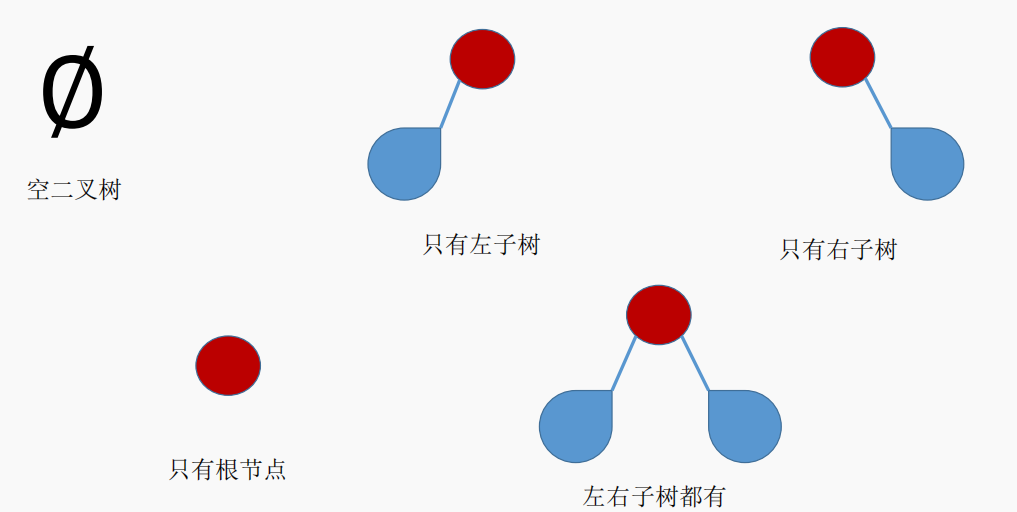
特点：①每个结点**至多只有两棵子树** ②左右子树**不能颠倒**（二叉树是**有序树**）

注意区别：度为2的有序树



二叉树是**递归**定义的数据结构

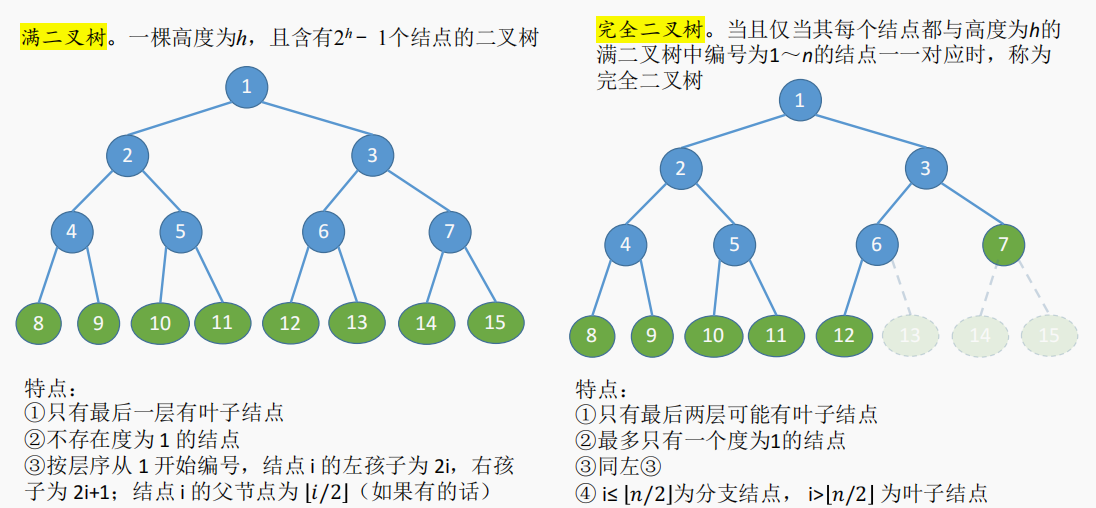
二叉树的五种状态:

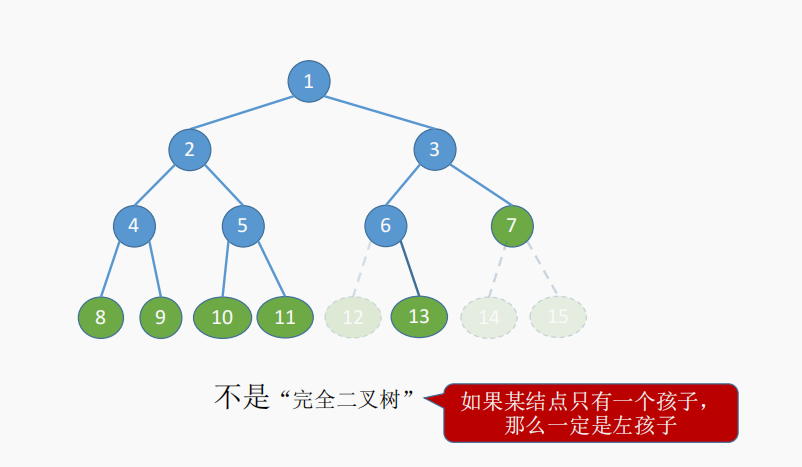


几个特殊的二叉树:

（1）**满二叉树**。一棵高度为h，且含有**2h - 1**个结点的二叉树。

（2）**完全二叉树**。当且仅当其**每个结点都与高度为h的满二叉树中编号为1～n的结点一一对应**时，称为完全二叉树。



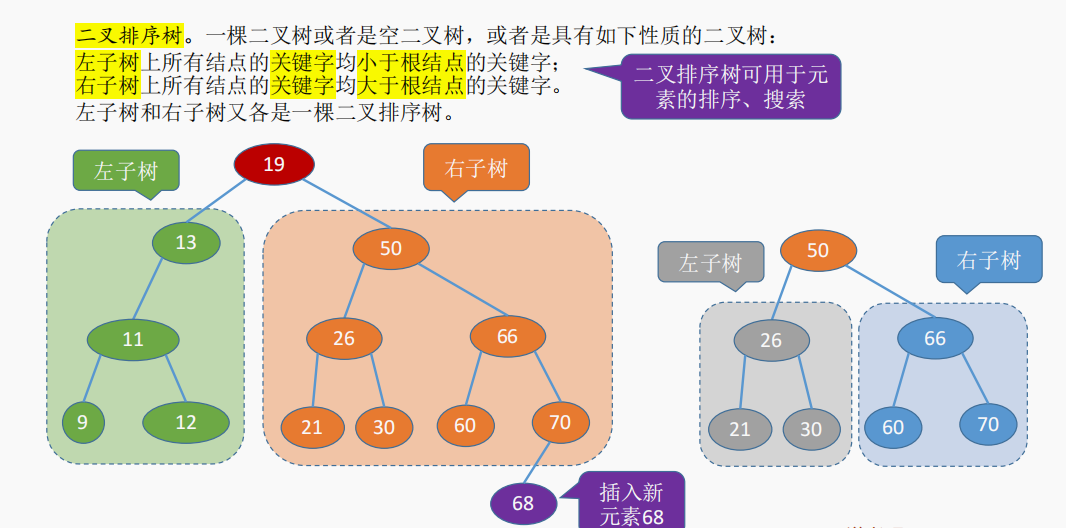


（3）**二叉排序树**。一棵二叉树或者是空二叉树，或者是具有如下性质的二叉树：

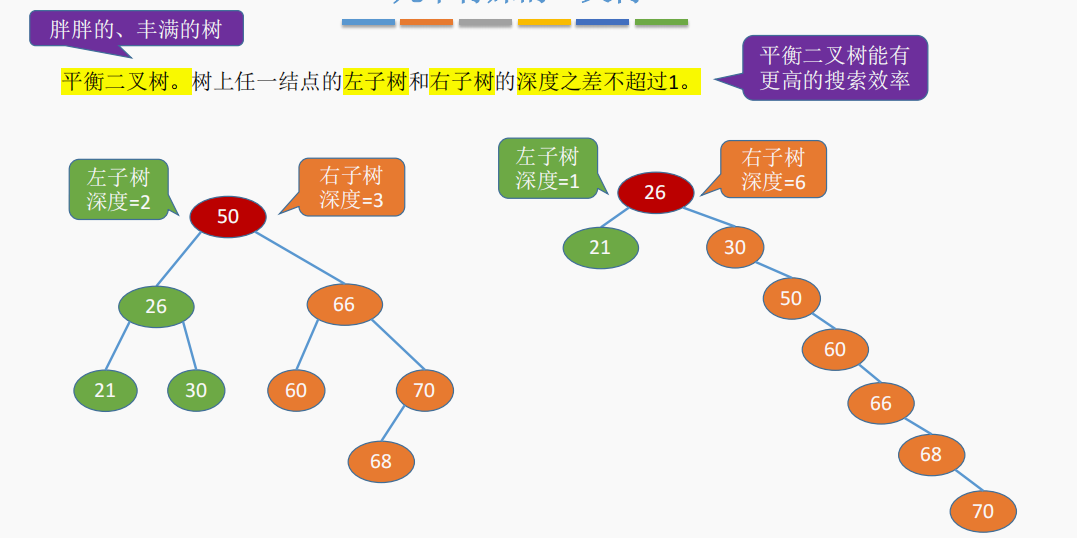
**左子树**上所有结点的**关键字**均**小于根结点**的关键字；

**右子树**上所有结点的**关键字**均**大于根结点**的关键字。

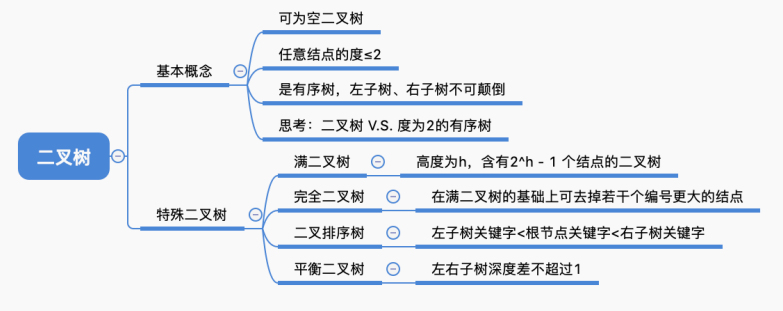
**左子树和右子树又各是一棵二叉排序树**。



（4）**平衡二叉树**。树上任一结点的**左子树和右子树的深度之差不超过1**。



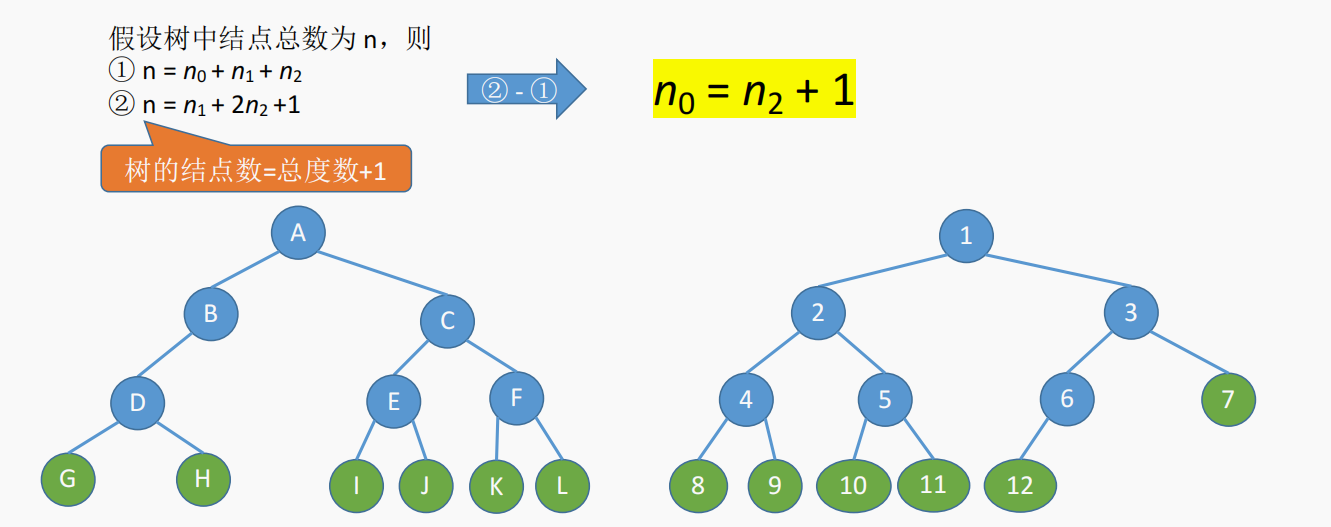
总结：



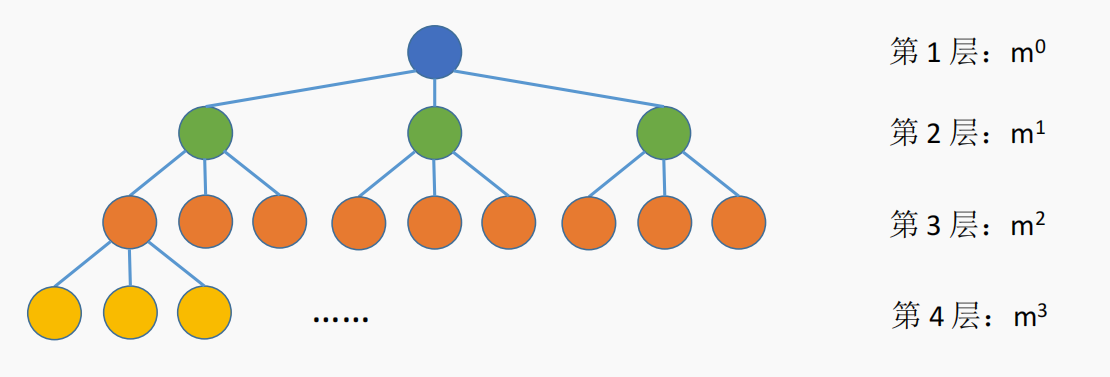
二叉树常考性质

常见考点1：设非空二叉树中度为0、1和2的结点个数分别为n0、n1和n2，则 **n0 = n2 + 1**

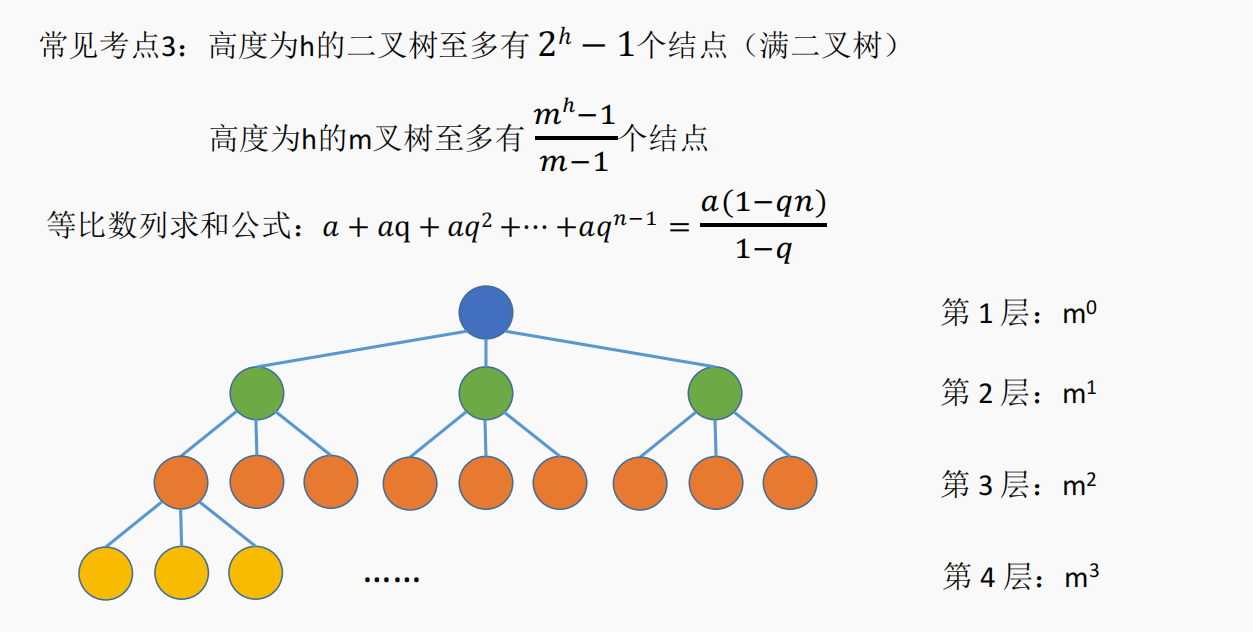
（叶子结点比二分支结点多一个）



常见考点2：二叉树第 i 层至多有 2**i-1**个结点（i≥1）

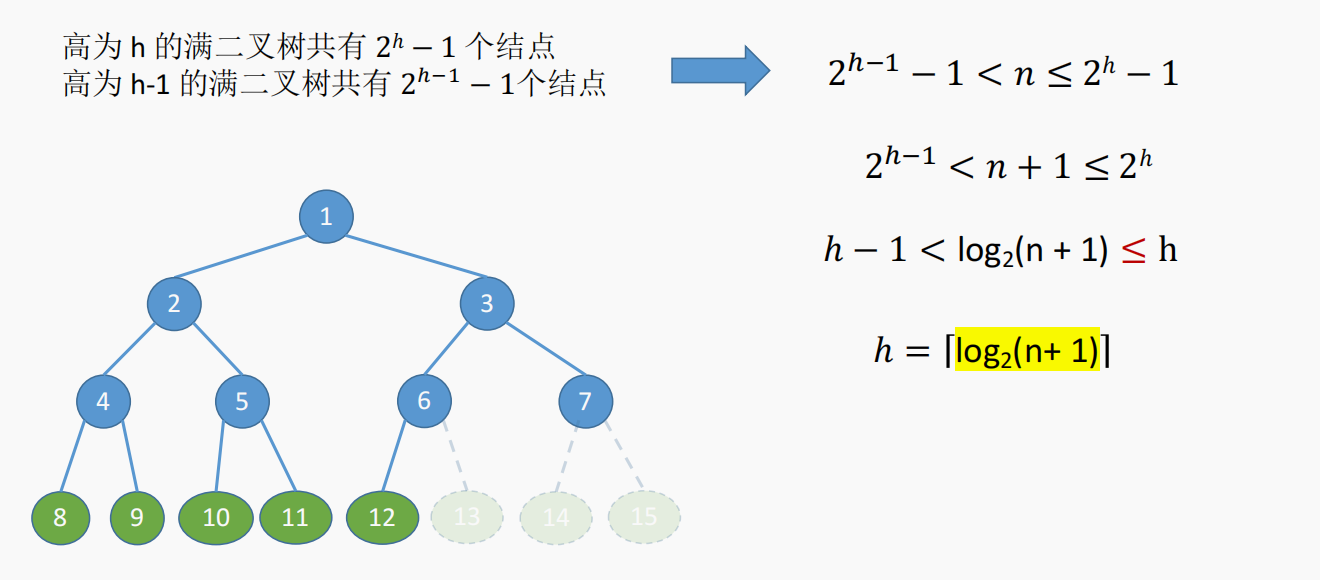
m叉树第 i 层至多有 mi-1 个结点（i≥1）

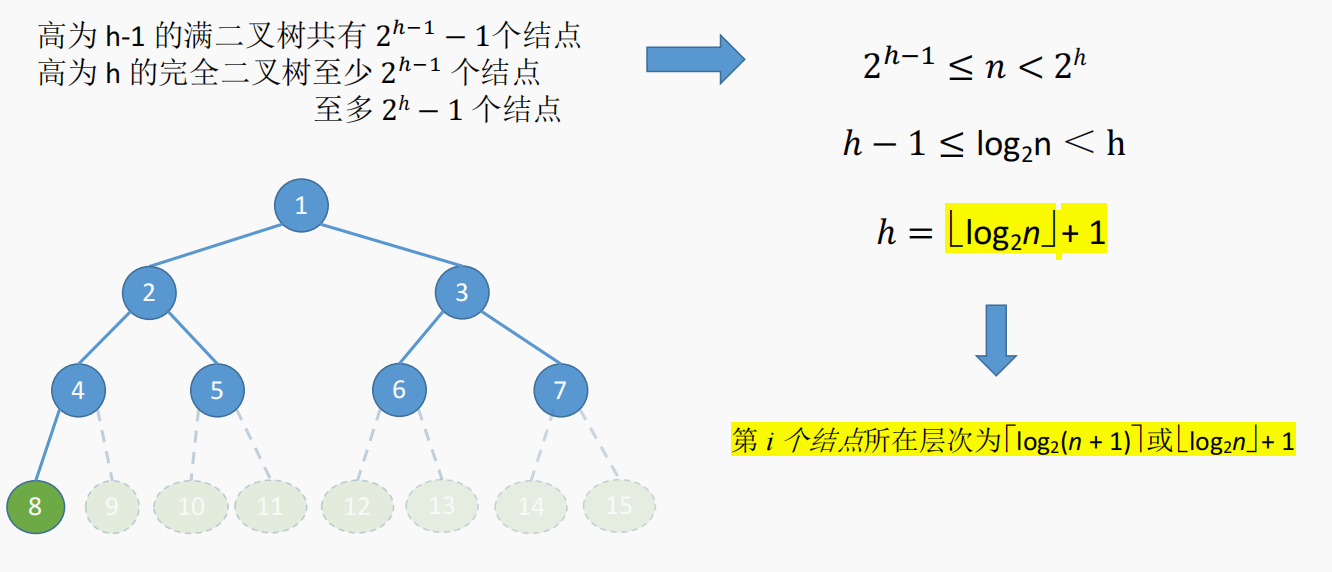
常见考点3：高度为h的二叉树至多有 2ℎ − 1个结点（满二叉树）



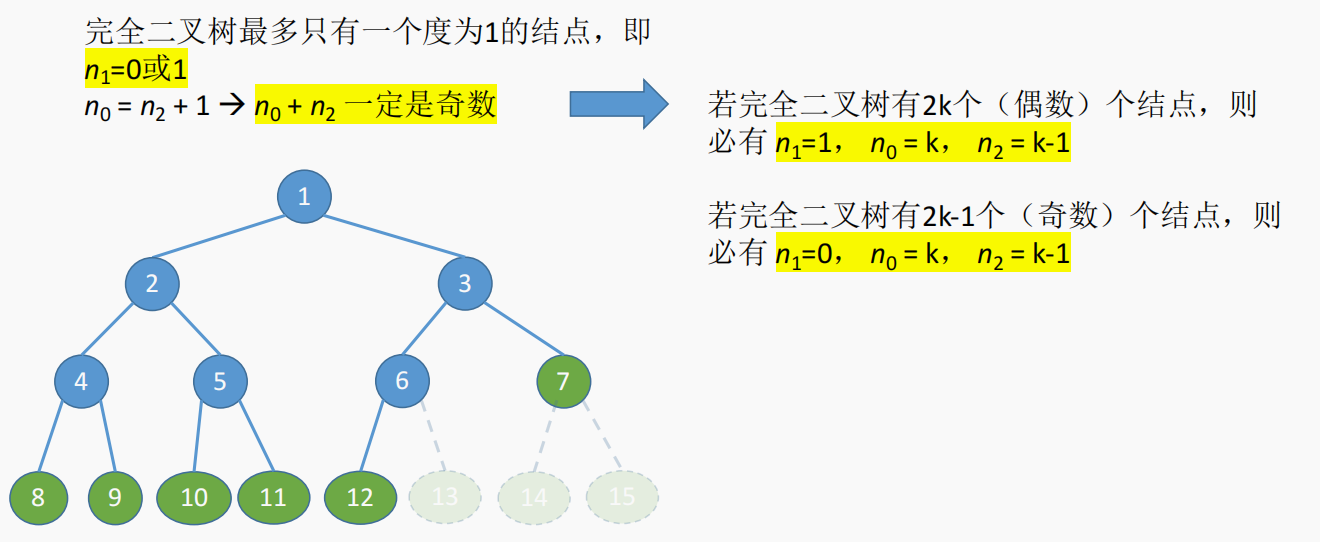
完全二叉树的常考性质：

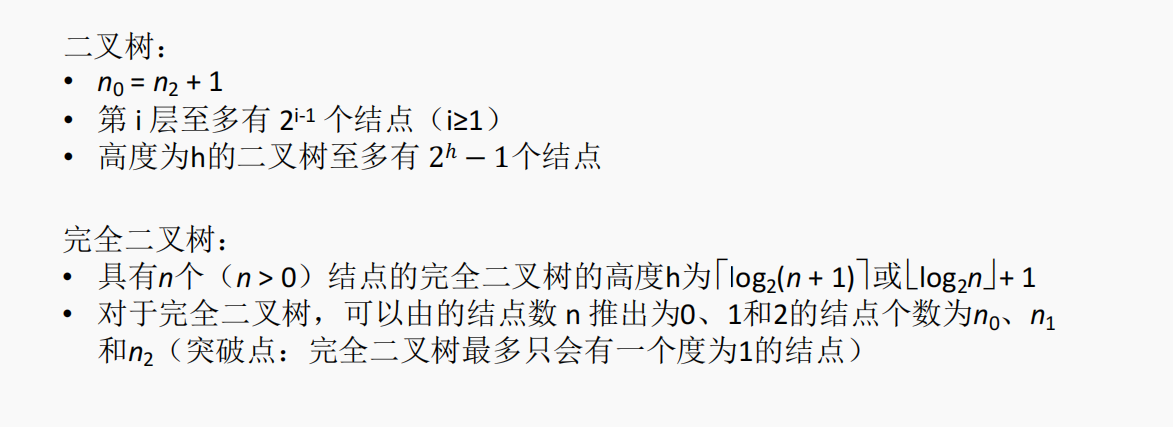
常见考点1：具有*n*个（*n* > 0）结点的完全二叉树的高度h为élog2(*n* + 1)ù或ëlog2*n*û + 1





常见考点2：对于完全二叉树，可以由的结点数 n 推出度为0、1和2的结点个数为n0、n1和n2。

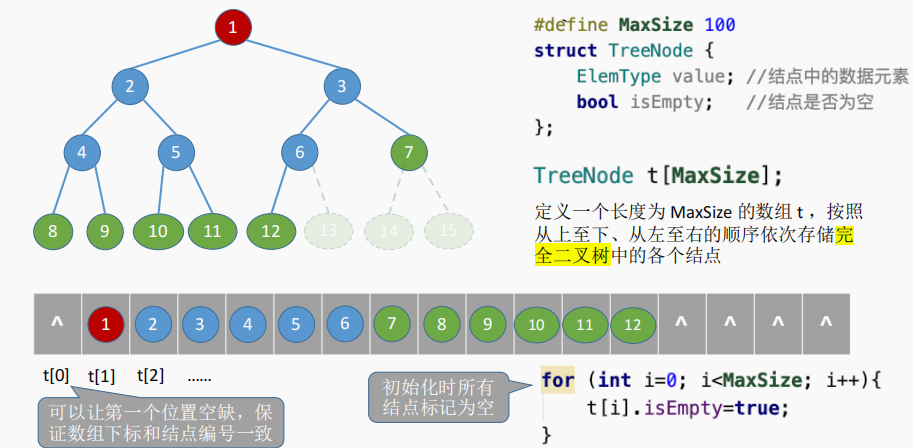


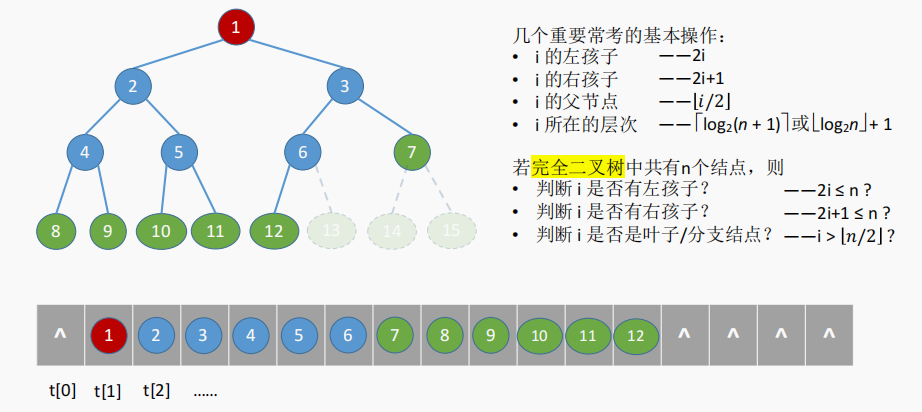


二叉树存储结构



二叉树的顺序存储：





几个重要常考的基本操作：

• i 的左孩子 ——2i

• i 的右孩子 ——2i+1

• i 的父节点 —— ëi/2û

• i 所在的层次 —— élog2(*n* + 1)ù或ëlog2*n*û + 1

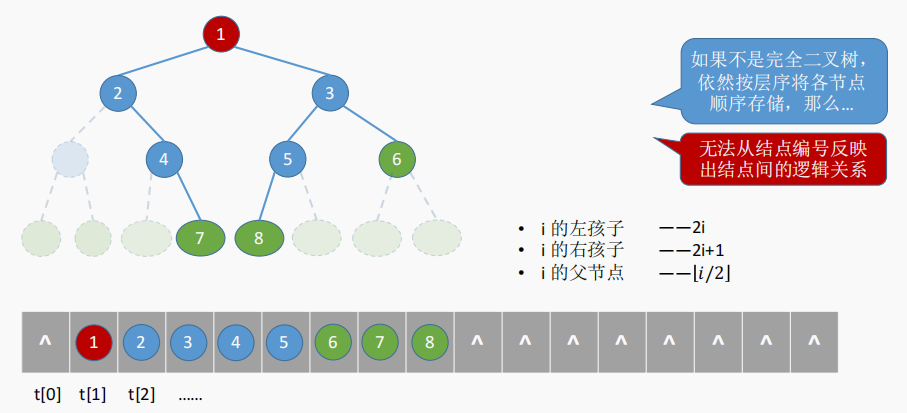
若完全二叉树中共有n个结点，则

• 判断 i 是否有左孩子？ ——2i ≤ n ?

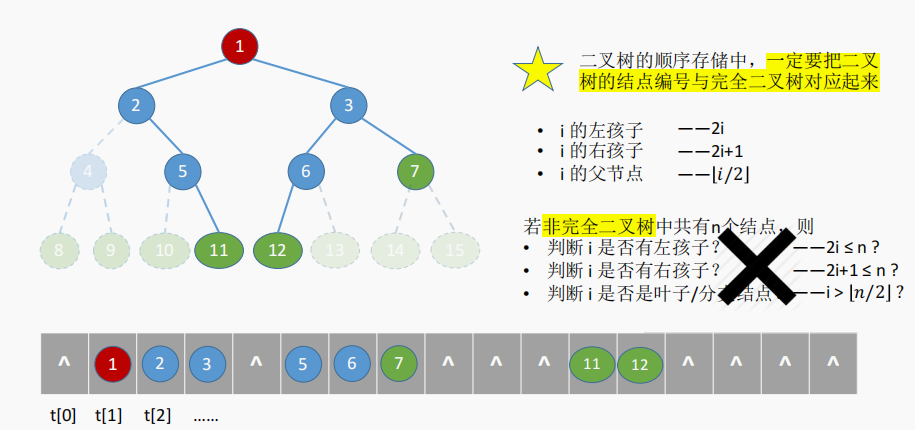
• 判断 i 是否有右孩子？ ——2i+1 ≤ n ?

• 判断 i 是否是叶子/分支结点？——i > ën/2û

如果**不是完全二叉树**，依然**按层序将各节点顺序存储**，那么**无法从结点编号反映出结点间的逻辑关系**

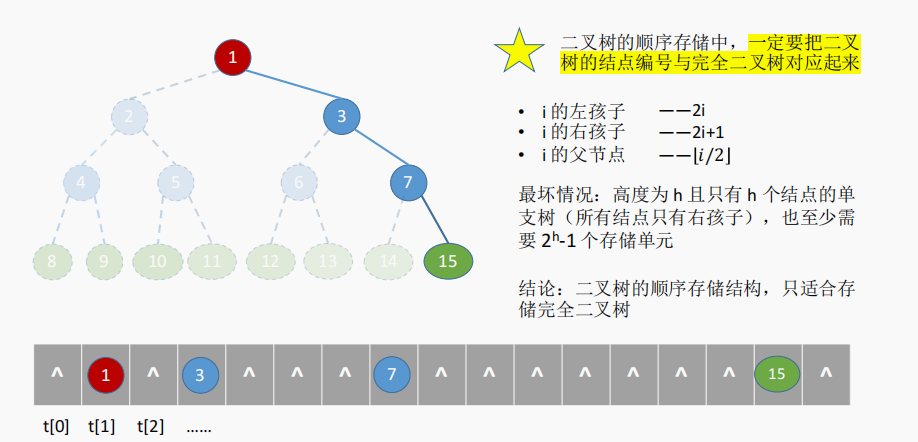


二叉树的顺序存储中，一**定要把二叉树的结点编号与完全二叉树对应起来**

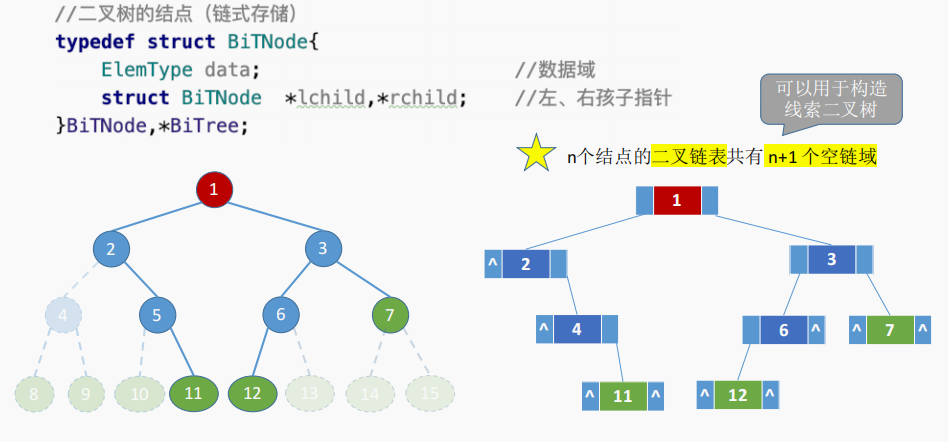


最坏情况：高度为 h 且只有 h 个结点的单支树（所有结点只有右孩子），也至少需要2h-1 个存储单元

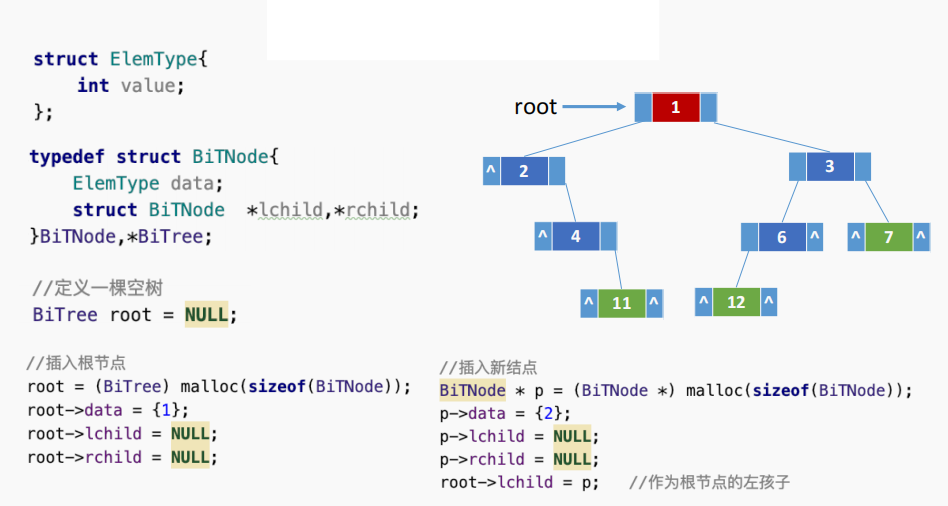
结论：二叉树的**顺序存储结构**，**只适合存储完全二叉树**



**二叉树的链式存储：**



**n个结点**的二叉链表共有 **n+1** 个空链域



**三叉链表：方便找父结点**

找到指定结点 p 的左/右孩子——超简单

如何找到指定结点 **p 的父结点**？**只能从根开始遍历寻找**

