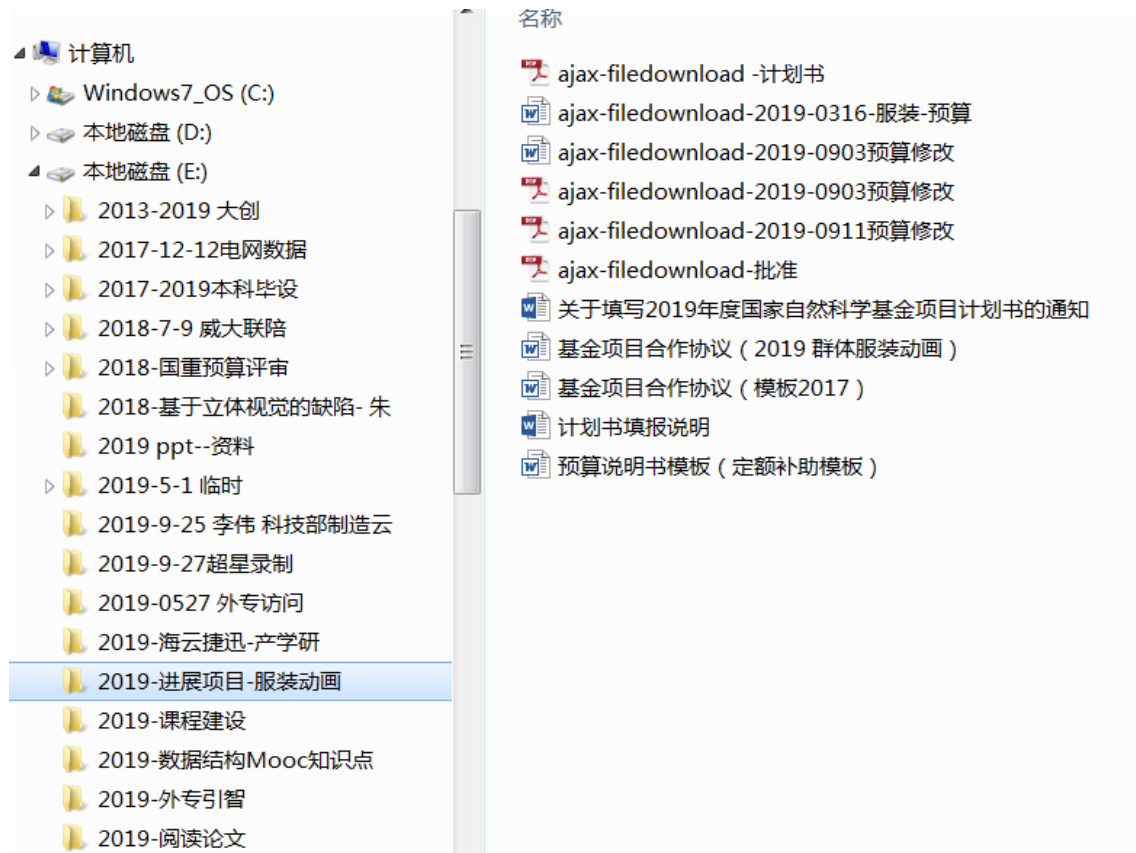
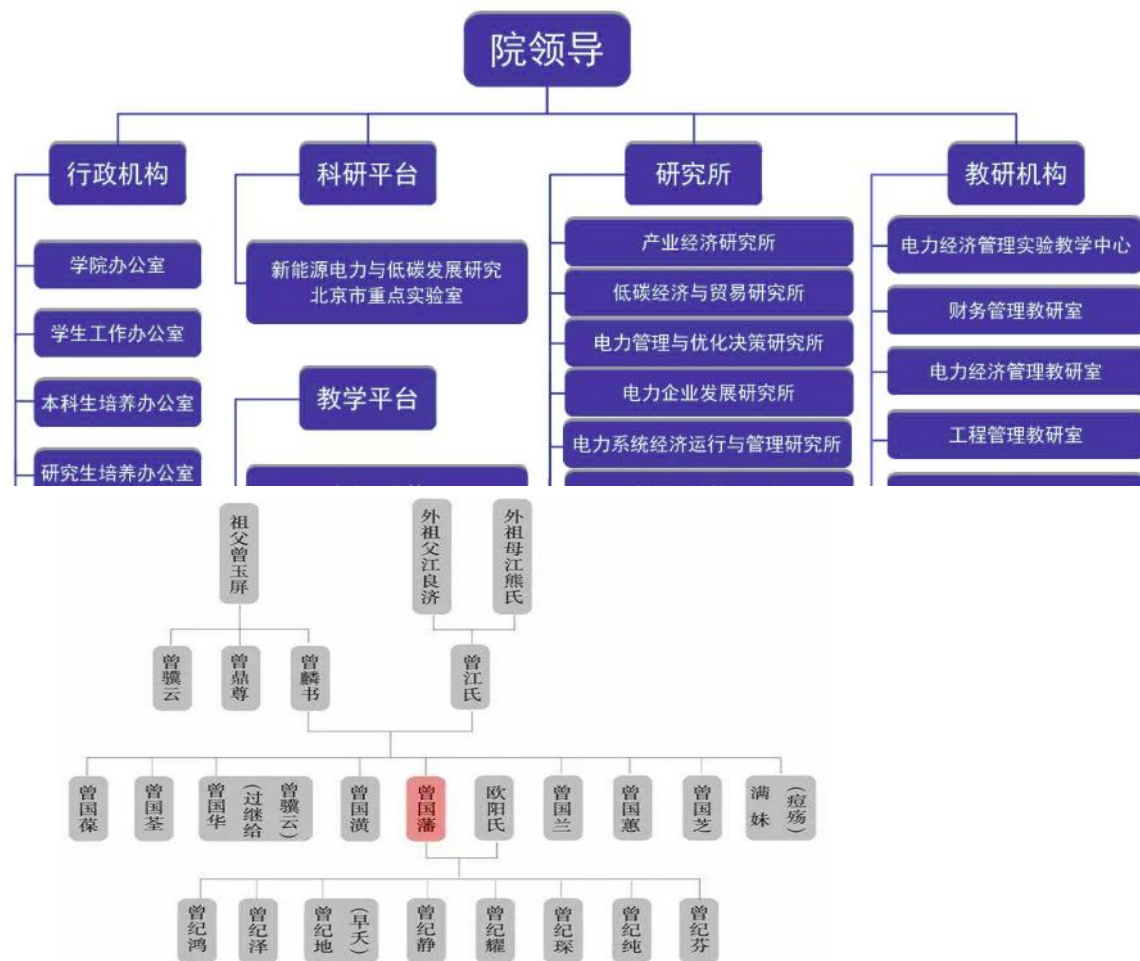


预习要点

- 什么是树
- 什么是二叉树
- 树和二叉树的不同
- 二叉树的5个基本性质
- 二叉树的两种存储结构

第5章 树与二叉树

思考以下问题



本章内容

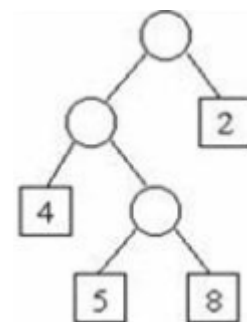
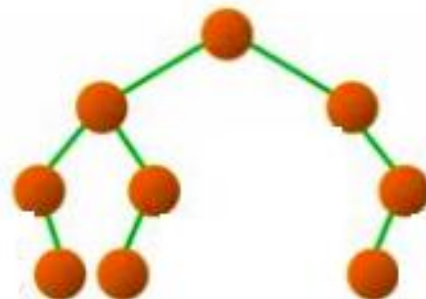
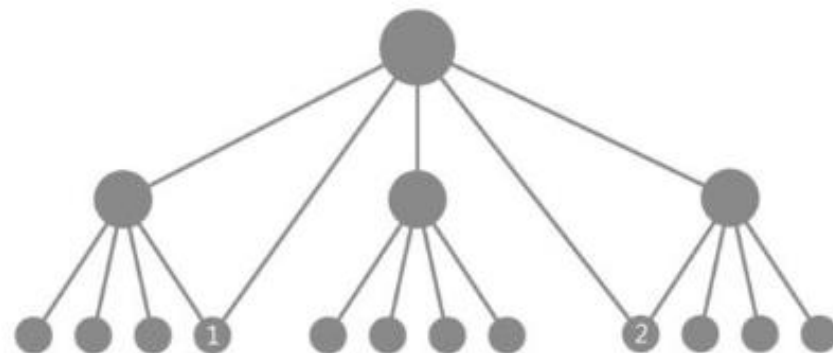
5.1 树的定义和基本术语

5.2 二叉树

5.3 遍历二叉树

5.4 树和森林的转换

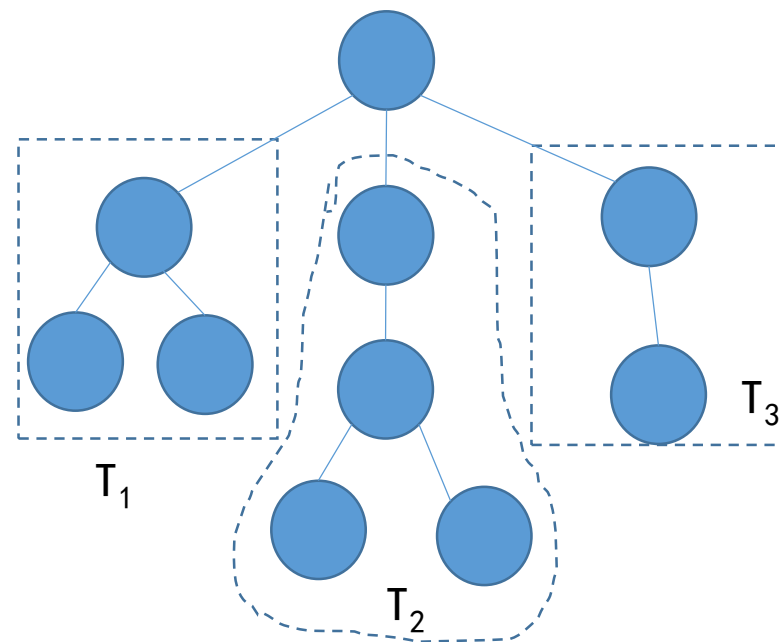
5.5 哈夫曼树及其应用



5.1 树的定义和基本术语

(1) 树的概念：是 n 个结点的有限集合。

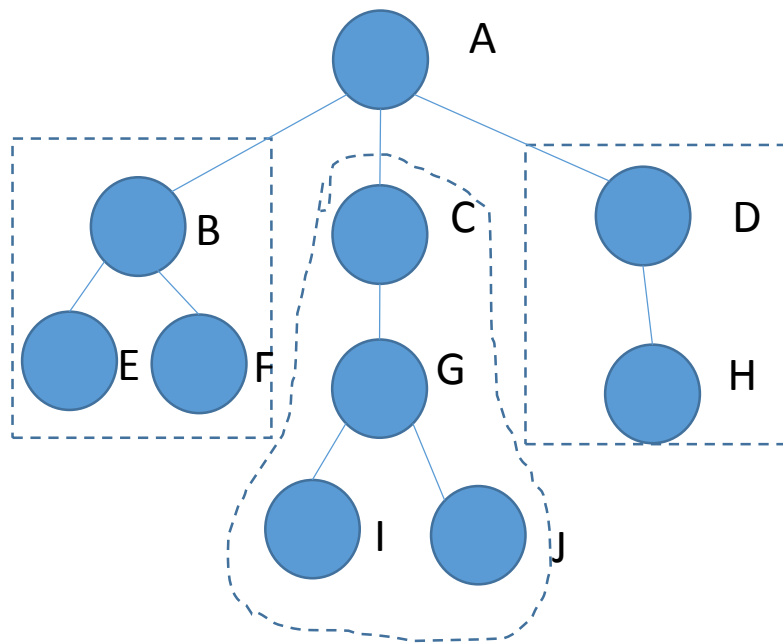
非空树 { 根结点
 m 个互不相交的有限集：
 T_1, T_2, \dots, T_m



其中每一个集合本身又是一棵树，并且称为根的子树。树的定义是递归的。

(2) 树的表示

第一种表示：结点连线。隐含：上方结点是下方结点的前驱。



A是根

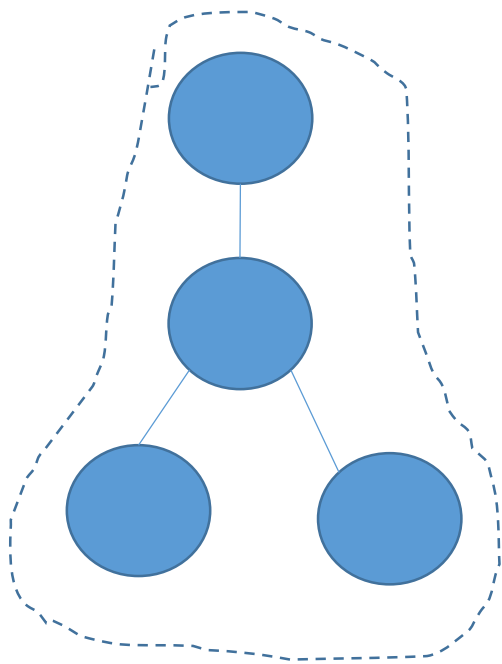
$T1 = \{B, E, F\}$

$T2 = \{C, G, I, J\}$

$T3 = \{D, H\}$

在树中，每个结点被定义为它的子树的根结点的前驱。

第二种表示：二元组表示。 如上面的T2可表示为：



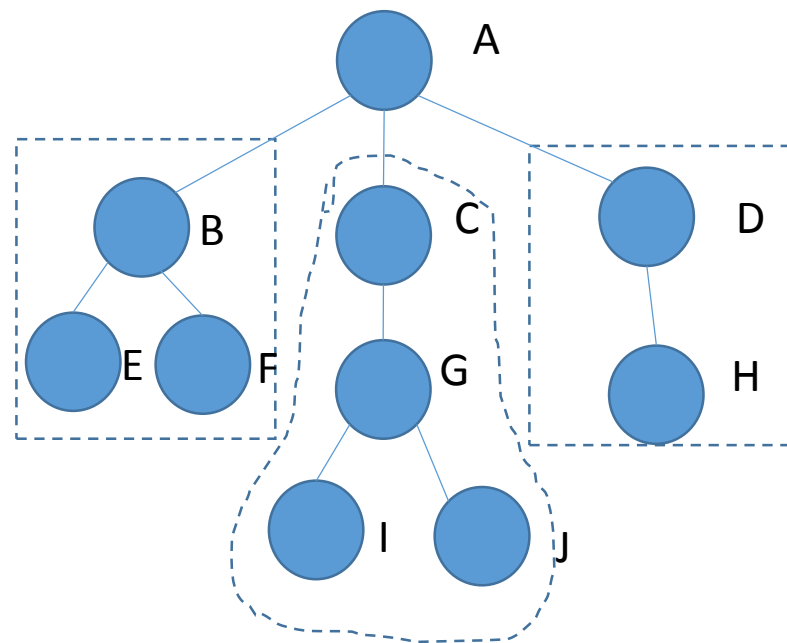
$$K=\{C,G,I,J\}$$

$$R=\{<C,G>,<G,I>,<G,J>\}$$

关系 r 应满足以下关系：

- 根结点没有前驱；
- 其余每个结点只有一个前驱结点
- 任何结点可以有多个后继

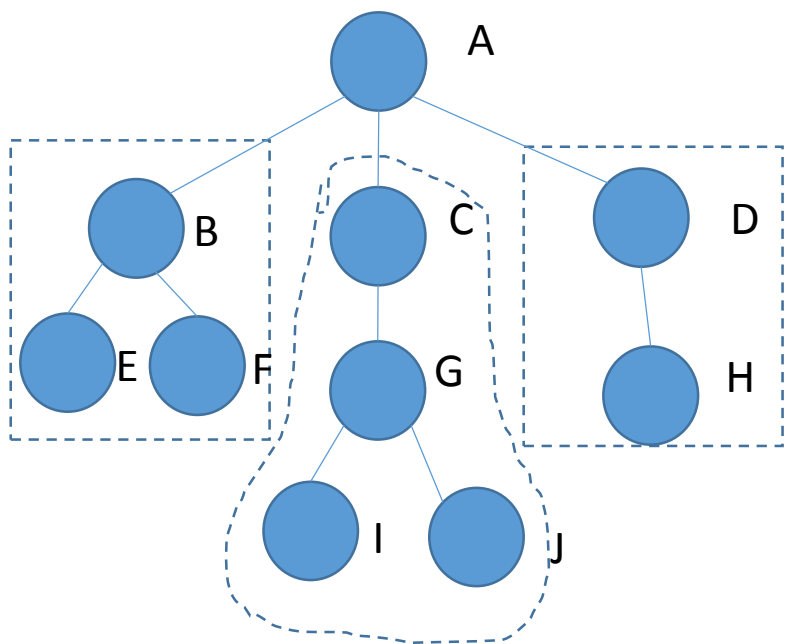
第三种：广义表表示



A(B(E, F), C(G(I, J)), D(H))

树根 T_1 T_2 T_3

(3) 基本术语



(1)树的结点：包含一个数据元素及若干指向其子树的分支

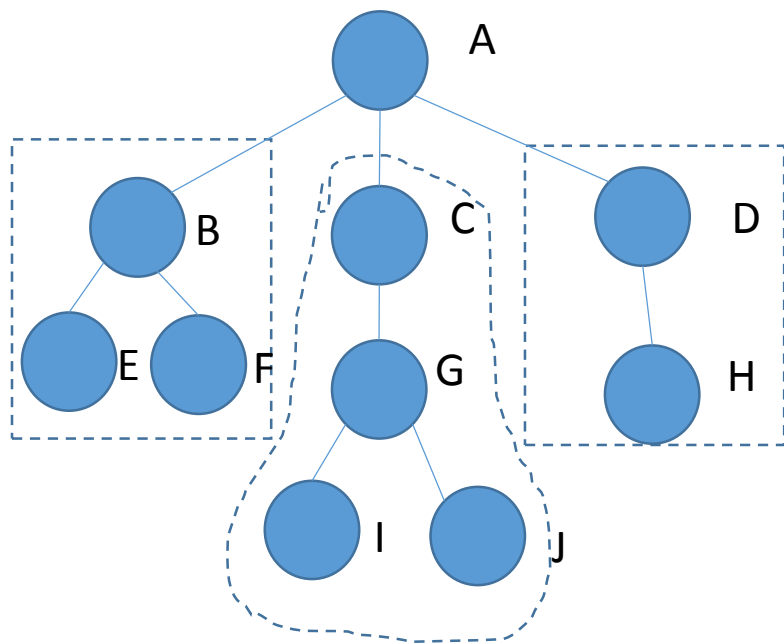
(2)结点的度（degree）：结点拥有的子树数。

(3)树的度：树内各结点的度的最大值。

(4)根结点：无前驱。

(5)叶子（终端结点）：度为零的结点。

(6)分支结点（非终端结点）：度不为零的结点。除根结点外，分支结点也称内部结点。



(7) 孩子结点、双亲结点、和兄弟结点：

(8) 堂兄弟：

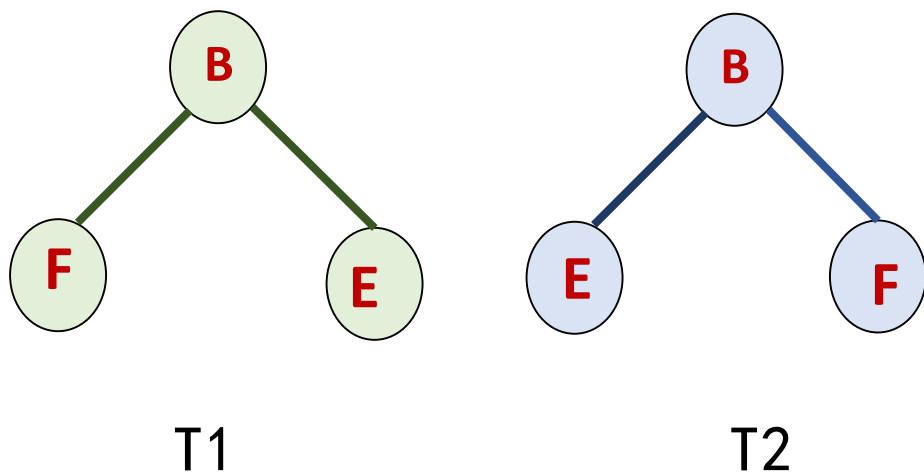
(9) 祖先：结点的祖先是根到该结点所经分支上的所有结点。

(10) 子孙：以某结点为根的子树中的任一结点都称为该结点的子孙。

(11) 结点的层次：根为第一层，根的孩子为第二层。

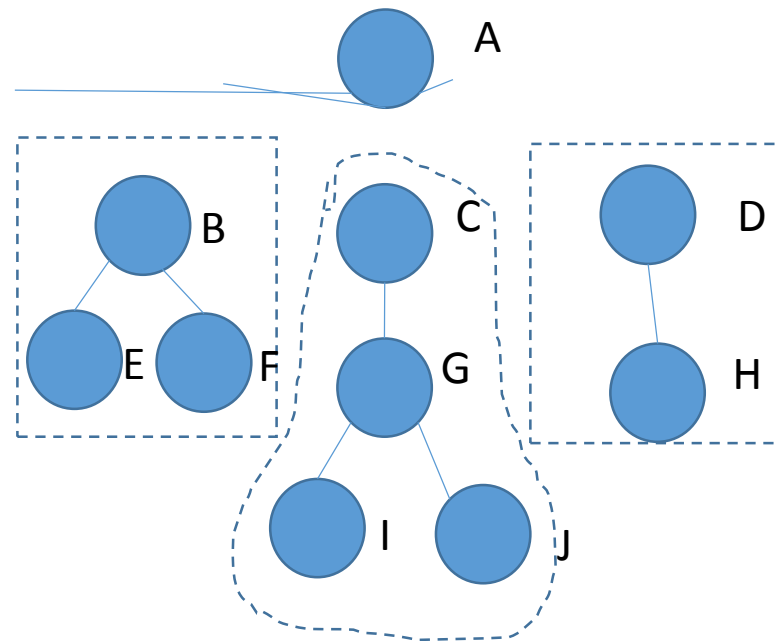
(12) 树的深度（高度）：树中结点的最大层次。

(13) 有序树及无序树



家族树通常是有序树

(14) 森林：M棵互不相交的树的集合



树结构与线性结构的比较

线性结构：

- 第一个数据元素 无前驱；
- 最后一个数据元素 无后继；
- 其它数据元素
一个前驱、一个后继

树型结构：

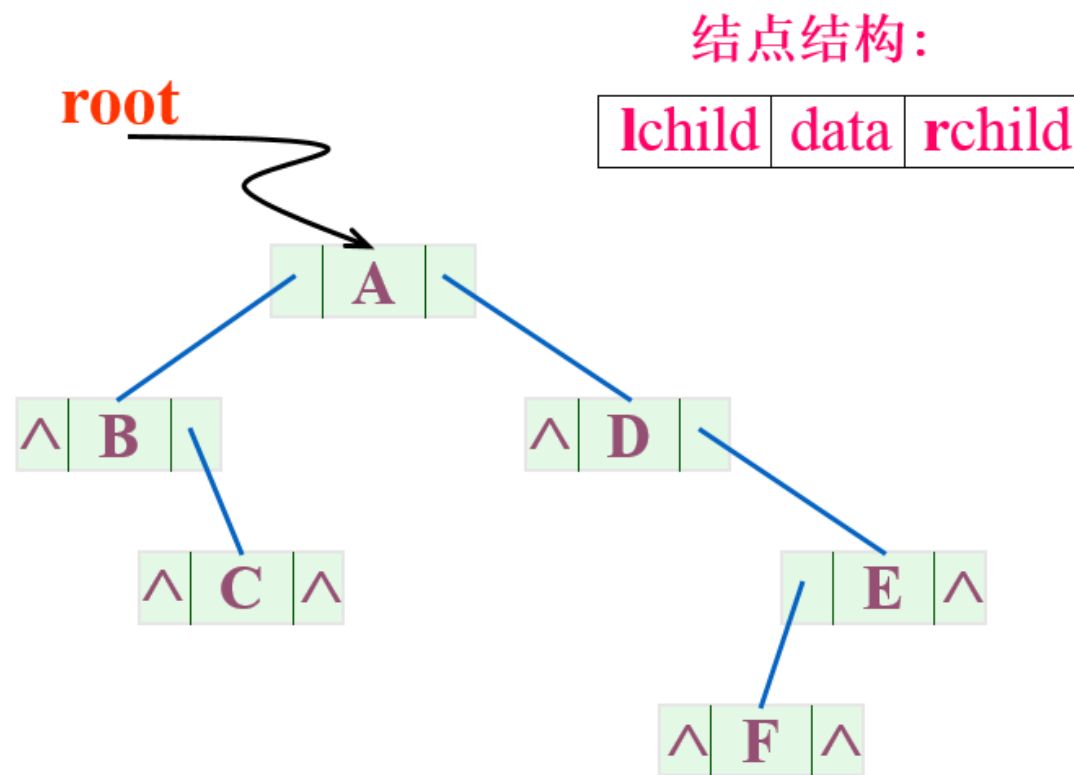
- 根结点无前驱
- 多个叶子结无后继
- 其它数据元素一个前驱、多个后继

5.2 二叉树

5.2.1 二叉树的定义和性质

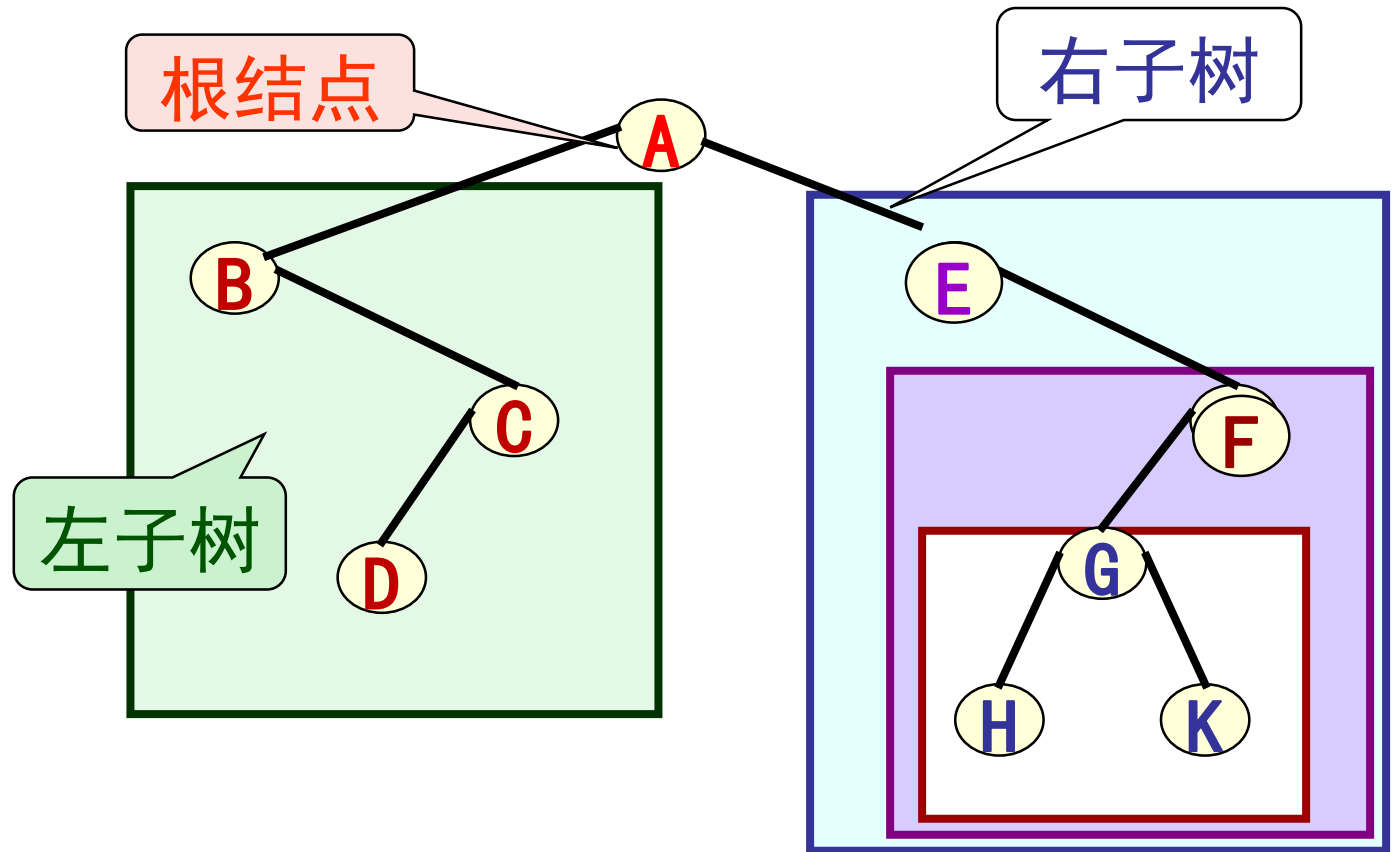
5.2.2 二叉树的存储结构

5.2.3 二叉树的运算



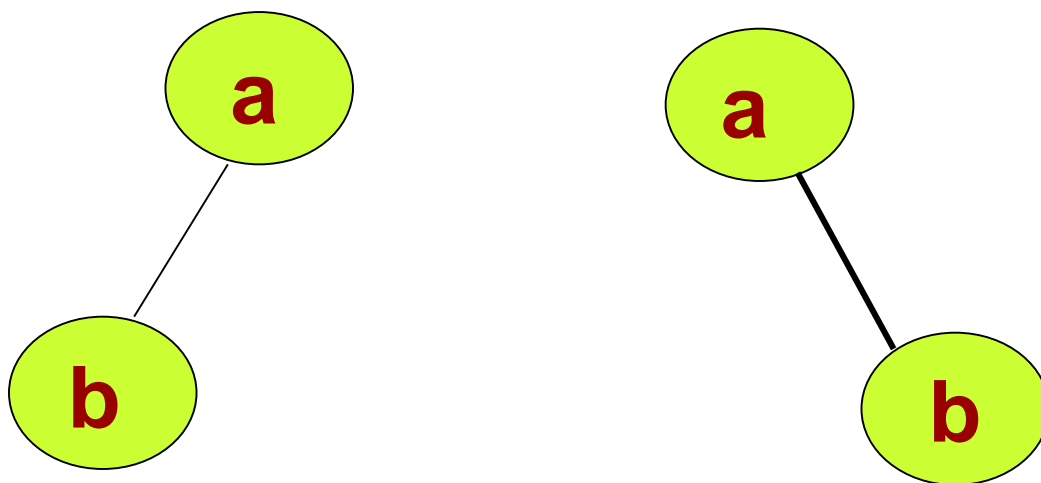
5.2.1 二叉树的定义及其性质

一、定义：二叉树是 n ($n \geq 0$) 个结点的有限集合。它或为空树 ($n=0$)，或由一个根结点和两棵分别称为根的左子树和右子树的互不相交的二叉树组成。



这也是一个递归定义。二叉树可以是空集合，根可以有空的左子树或空的右子树。

特点： 每个结点至多只有两棵子树，子树有左右之分，其次序不能任意颠倒，分别称为左右子树。

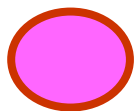


两棵不同的二叉树

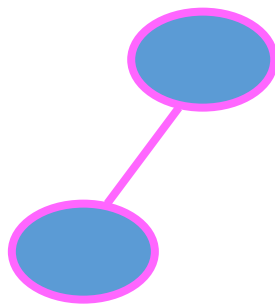
二叉树的五种基本形态



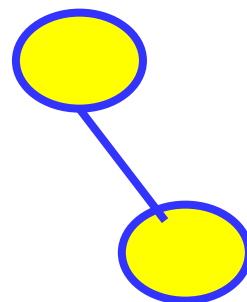
空二叉树



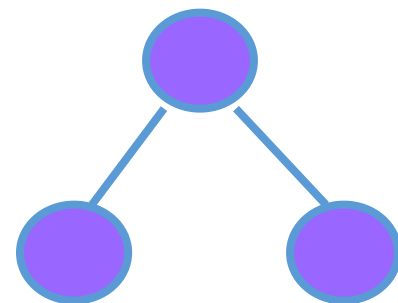
仅有
根结点



右子树
为空



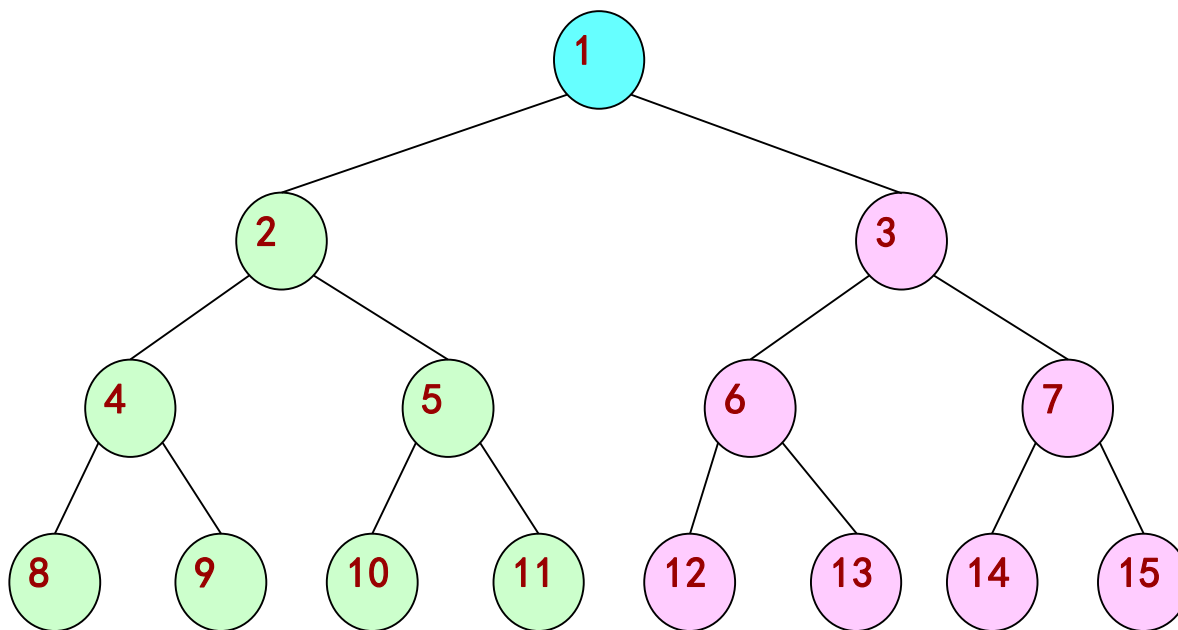
左子树
为空



左右子树均
非空

二、特殊的二叉树

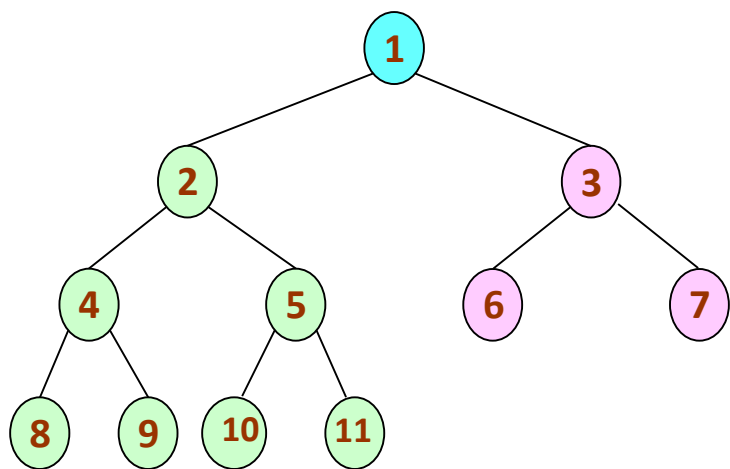
(1) 满二叉树 深度为 k 且含有 2^k-1 个结点的二叉树



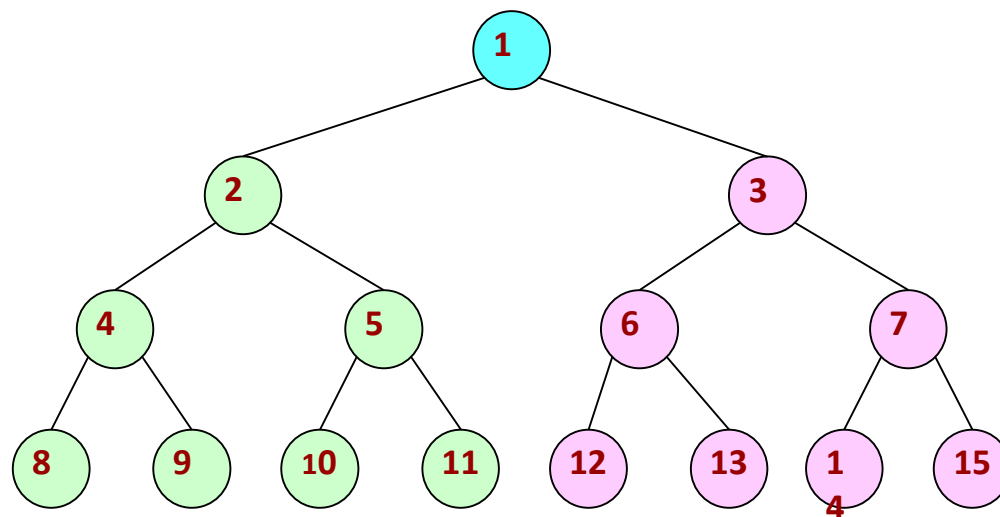
特点：每一层上都含有最大结点数。

(2) 完全二叉树

树中所含的 n 个结点和满二叉树中编号为 1 至 n 的结点一一对应。



完全二叉树



满二叉树

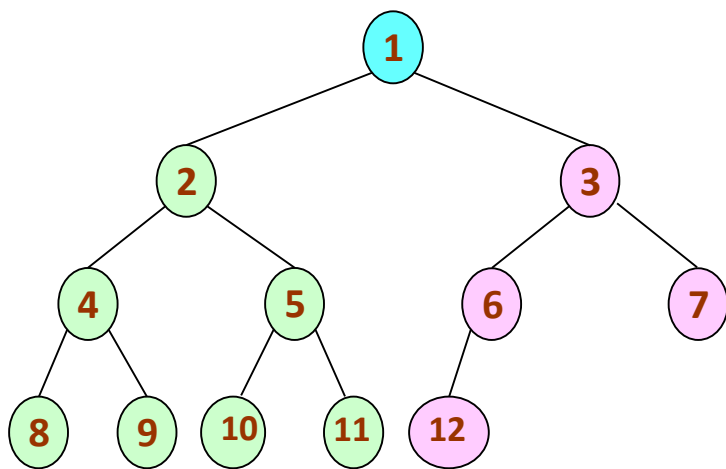
满二叉树是完全二叉树的特例

特点：

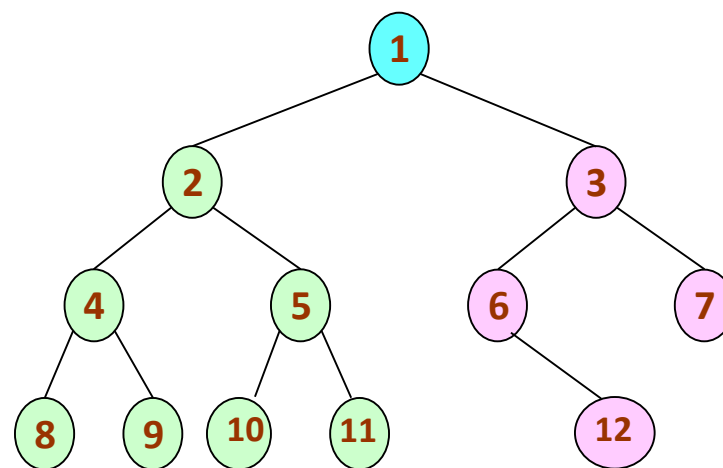
- 除最后一层外，每一层都取最大结点数
- 深度为 k 的完全二叉树，所有的叶结点都出现在第 k 层或 $k-1$ 层
- 最后一层结点都集中在该层最左边的若干位置
- 如果其右子树的最大层次为 L ，则其左子树的最大层次为 L 或 $L+1$.

(3) 理想平衡树

在一棵二叉树中，若除最后一层外，其余层都是满的，则称此树是理想平衡树。理想平衡树包括满二叉树和完全二叉树。但并不一定是完全二叉树。



完全二叉树



理想平衡树

三、 二叉树的五个基本性质

性质1: 在二叉树的第 k 层上至多有 2^{k-1} 个结点 ($k \geq 1$)

用归纳法证明:

(1) $k=1$ 时, $2^{k-1}=1$, 即只有一个根结点。

(2) 设对 $1 \leq j \leq k-1$, 命题成立, 则可以证明 $j=k$ 时命题也成立。

由于第 $k-1$ 层上至多有 2^{k-2} 个结点, 由于二叉树中每个结点的度至多为2, 故第 k 层上的结点数最多为 $2^{k-2} \times 2 = 2^{k-1}$ 。

性质2：深度为k的二叉树至多有 2^k-1 个结点 ($k \geq 1$)

$$\sum_{i=1}^k \text{第} i \text{层上的最大结点数} = \sum_{i=1}^k 2^{i-1} = (1+2+2^2+\cdots+2^{k-1}) = 2^k-1$$

性质3：对任何一棵二叉树T，如果其终端结点数为 n_0 ，度为2的结点数为 n_2 ，则有 $n_0 = n_2 + 1$ 。

证明 (3) : 设度为 1 的结点数为 n_1 , 二叉树的总结点数为 n , 则有

$$n = n_0 + n_1 + n_2;$$

整棵二叉树的分支数为 B , 由于除根结点外, 每一个结点都有一个分支到达, 则有

$$n = B + 1;$$

又由于这些分支数是由度为 1 和度为 2 的结点发出的, 则有:

$$B = n_1 + 2n_2。$$

$$\text{由上可得: } n_0 + n_1 + n_2 = n_1 + 2n_2 + 1$$

$$\text{即可导出: } n_0 = n_2 + 1$$

性质4: 具有 n 个结点的完全二叉树的深度为 $\lceil \log_2(n+1) \rceil$ 或 $\lfloor \log_2 n \rfloor + 1$ 。

假设深度为 k , 由性质2, 最大结点数为 2^k-1 , 深度为 $k-1$ 的最大结点数为 $2^{k-1}-1$ 。

$$2^{k-1}-1 < n \leq 2^k-1 \quad \text{或}$$

$$\downarrow$$
$$2^{k-1} < n+1 \leq 2^k$$

$$k-1 < \log_2(n+1) \leq k$$

$$\log_2(n+1) \leq k < \log_2(n+1) + 1$$

$$k = \lceil \log_2(n+1) \rceil$$

$$2^{k-1} \leq n < 2^k$$



$$k-1 \leq \log_2 n < k$$

$$k = \lfloor \log_2 n \rfloor + 1$$

性质5: 如果对一棵有 n 个结点的完全二叉树, 其深度为 $\lfloor \log_2 n \rfloor + 1$ 的结点按层序编号, 则对任一结点 i ($1 \leq i \leq n$), 有

(1) 如果 $i = 1$, 则结点 i 是根。如果 $i > 1$, 则其双亲 $\text{parent}(i)$ 是结点 $\lfloor i/2 \rfloor$ 。

(2) 如果 $2i > n$, 则结点 i 为叶子, 否则其左孩子 $\text{Lchild}(i)$ 是结点 $2i$ 。

(3) 如果 $2i+1 > n$, 则结点 i 无右孩子, 否则其右孩子是结点 $2i+1$ 。

证明(2), (3):

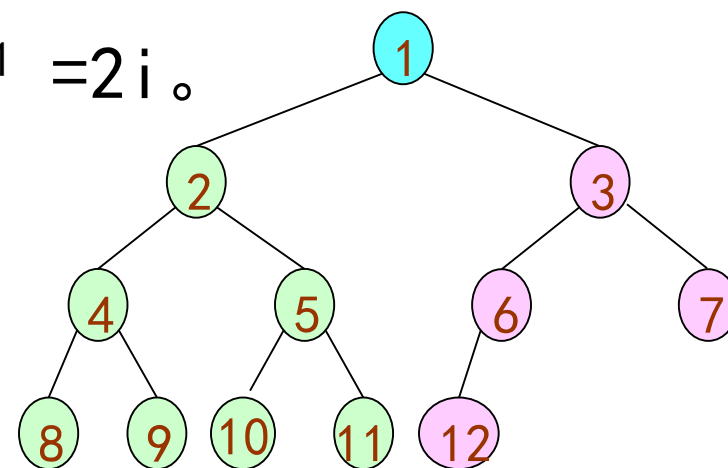
对于 $i=1$, 左孩子是2, 右孩子是3。

对于 $i>1$, 可分两种情况讨论:

第一种情况: 设第 j 层的第1个结点的编号为 i ($i=2^{j-1}$), 则其左孩子必为第 $j+1$ 层的第1个结点, 其编号为 $2^j = 2 * 2^{j-1} = 2i$ 。

若 $2i > n$, 则 i 无左孩子。右孩子编号为 $2i+1$,

若 $2i+1 > n$, 则无右孩子



完全二叉树

第二种情况：假设第j层上某个结点的编号为i，且 $2i+1 < n$ ，则其左孩子为 $2i$ ，右孩子为 $2i+1$ 。

则编号为 $i+1$ 的结点是编号为i的结点的右兄弟或堂兄弟，若它有左孩子，其编号必为 $2i+2=2(i+1)$ ，若它有右孩子，其编号必为 $2i+3=2(i+1)+1$ 。

由上可知，对于二叉树的任意一个结点，（2）（3）都成立

练习

练1. 具有 n 个结点的满二叉树，其叶结点的个数为多少？

练2. 设高为 h 的二叉树只有度为0和2的结点，则此类二叉树的结点数至少为？至多为？

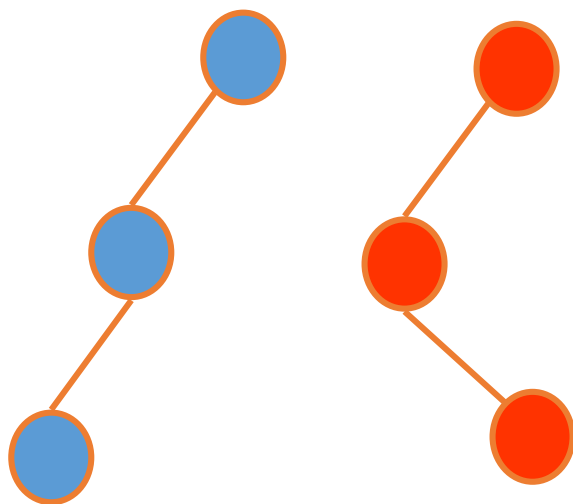
练3. 已知一棵度为 k 的树中有 n_1 个度为1的结点， n_2 个度为2的结点， n_3 个度为3的结点， n_k 个度为 k 的结点，问该树中多少个叶子结点？

练4. 已知一棵含有 n 个结点的树中，只有度为0和度为 k 的结点，求叶子结点数目？

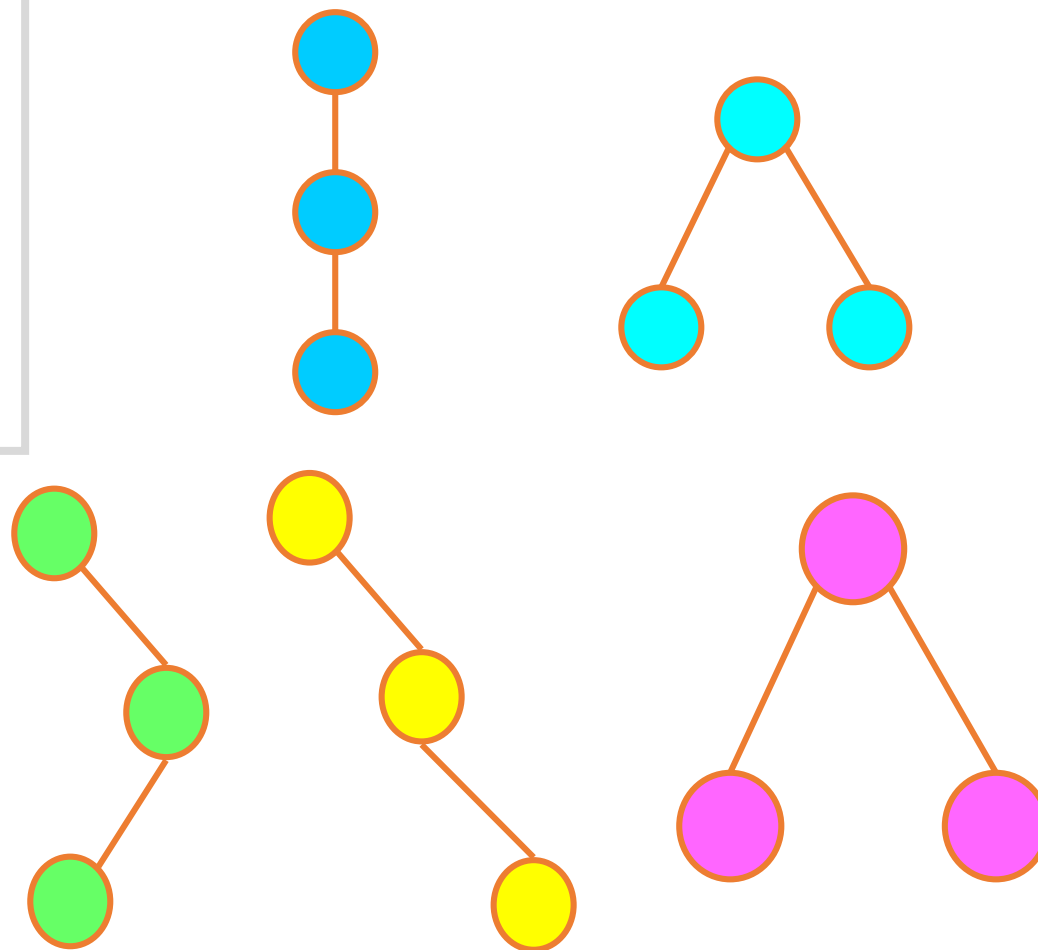
四、树与二叉树的区别

- 1) 树中结点的最大度数没有限制，二叉树结点最大度数为2。
- 2) 无明确指出，树没有左、右子树之分，二叉树有明确的左、右子树之分。

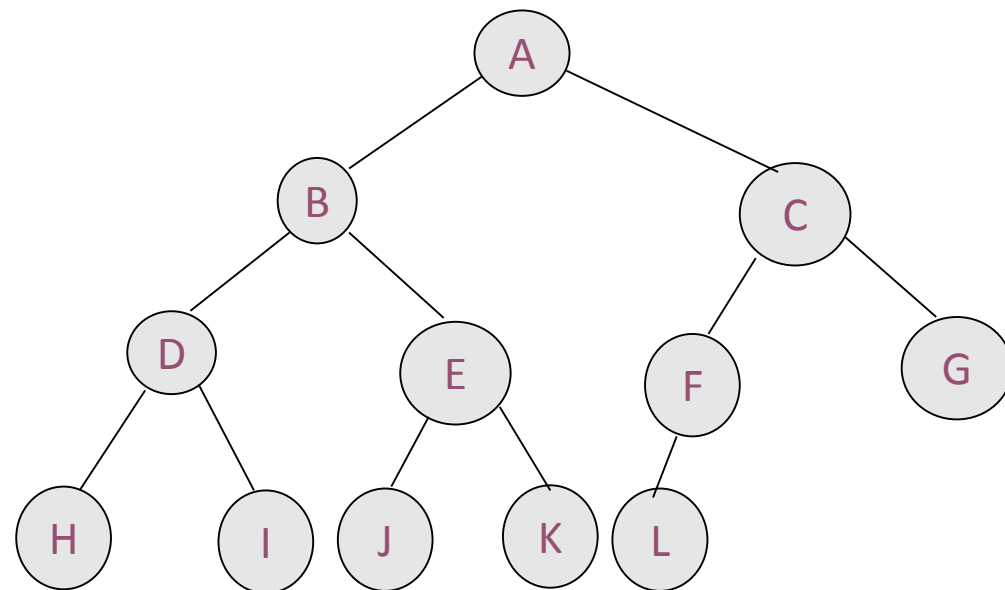
包含3个结点的
二叉树



包含3个结点的树



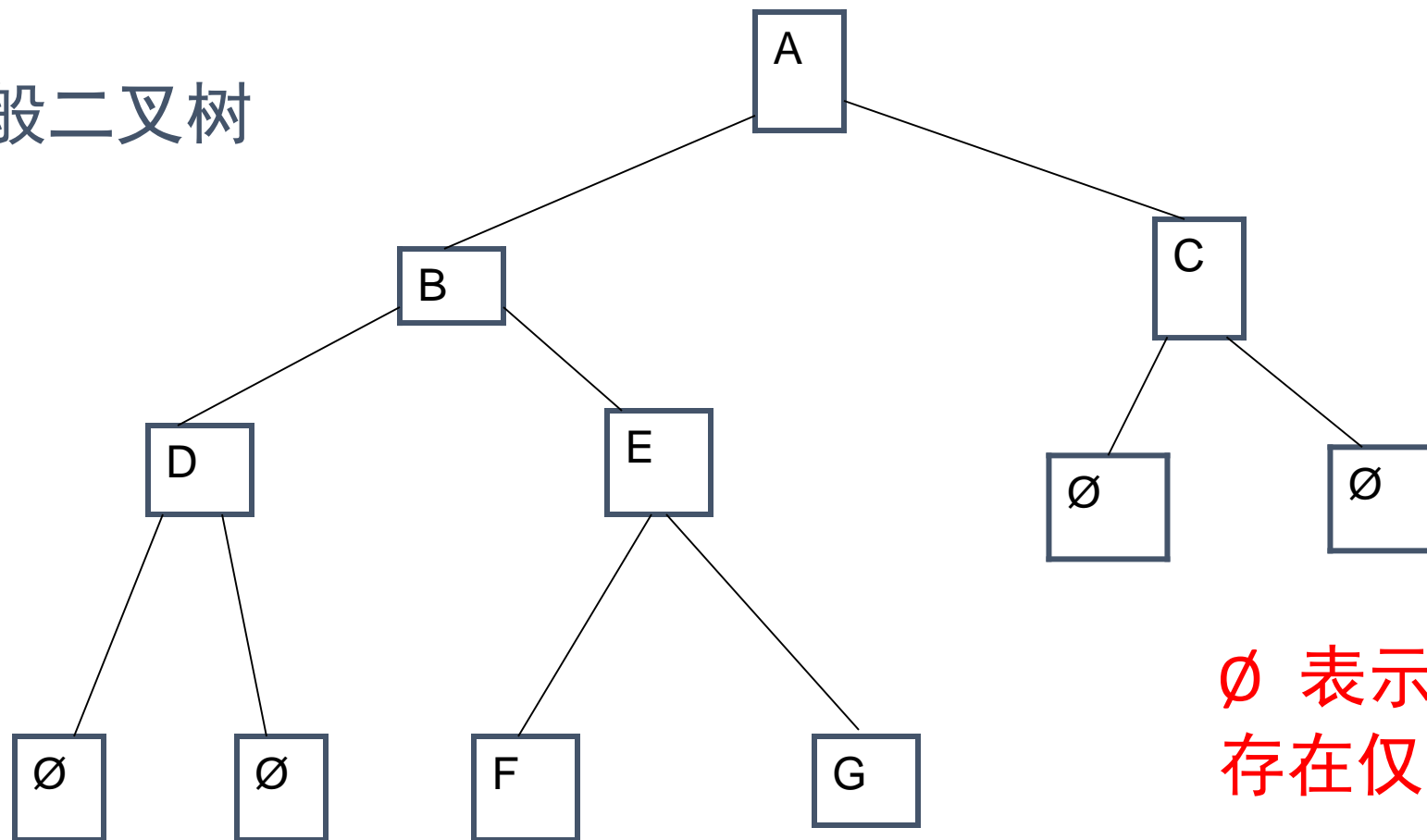
完全二叉树



	1	2	3	4	5	6	7	8	9	10	11	12
A[13]	A	B	C	D	E	F	G	H	I	J	K	L

若父结点在数组中 i 下标处，其左孩子在 $2*i$ ，右孩子在 $2*i+1$

一般二叉树

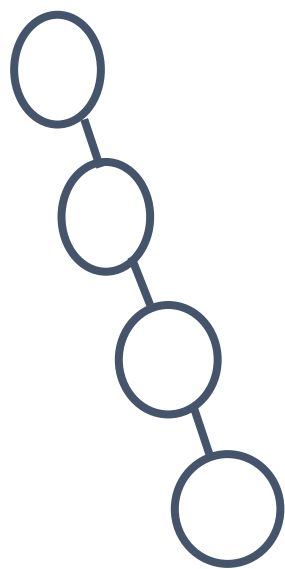


∅ 表示该处没有元素
存在仅仅为了好理解

1	2	3	4	5	6	7	8	9	10	11	
	A	B	C	D	E	∅	∅	∅	∅	F	G

若父结点在数组中 i 下标处，其左孩子在 $2*i$ ，右孩子在 $2*i+1$

问题：一个深度为H且只有H个结点的右单支树需要 2^h-1 个结点存储空间

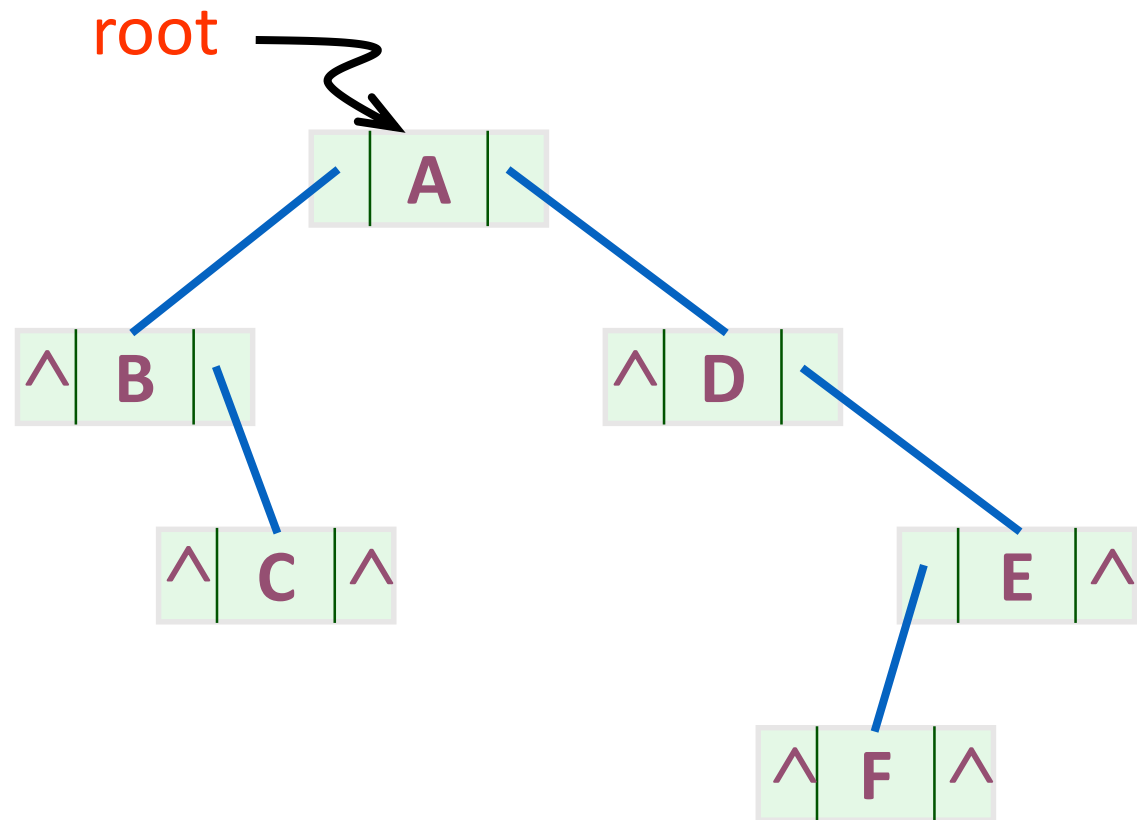


一般二叉树必须按完全二叉树的形式存储，将造成存储的浪费。

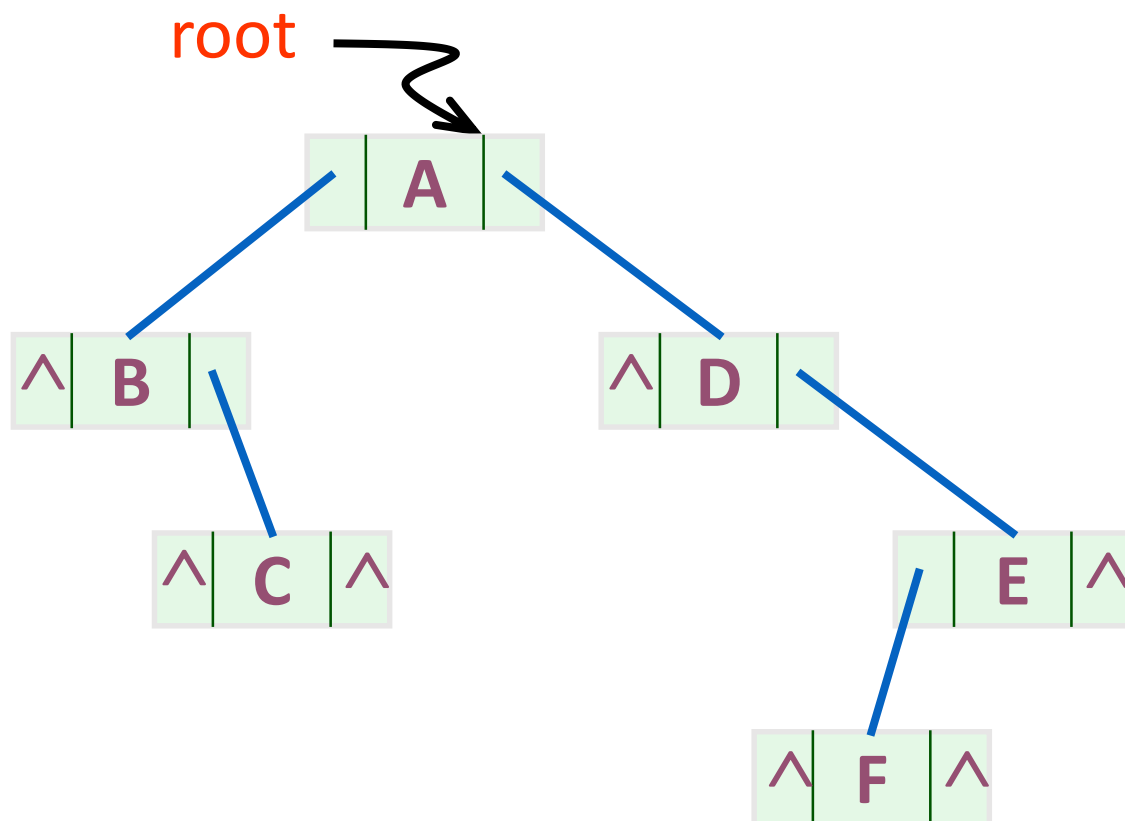
(2) 链式存储结构

根据结点结构不同

- 二叉链表
- 三叉链表



1. 二叉链表



结点结构:

lchild	data	rchild
--------	------	--------

```
typedef struct BiTNode {  
    ElemType    data;  
    struct BiTNode *lchild, *rchild;  
    // 左右孩子指针  
} BiTNode, *BiTree;
```

2. 三叉链表

结点结构:

parent	lchild	data	rchild
--------	--------	------	--------

```
typedef struct TriTNode {  
    ElemType    data;  
    struct TriTNode *lchild, *rchild;  
    // 左右孩子指针  
    struct TriTNode *parent;  
    // 双亲指针  
} TriTNode, *TriTree;
```

