第六章 树和二叉树

6.1 树的定义和基本术语

6.2 二叉树

6.3 遍历二叉树和线索二叉树

6.4 树和森林

6.6 赫夫曼树及其应用

6.1 树的定义和二叉树

一. 树的定义

- 树是有n(n≥0)个结点的有限集合。
  - 如果 n=0，称为空树
  - 如果 n≥1,称为非空树,对于非空树,有且仅有一个特定的称为根(Root)的结点,它没有前驱；除根以外的其它结点划分为 m (m ≥ 0)个互不相交的有限集 T1, T2 ,…, Tm，其中每个集合本身又是一棵树，并且称为根的子树(SubTree)。

✓ 定义说明：树是一种递归的数据结构——树中包含树。每棵子树的根结点有且仅有一个直接前驱，但可以有0个或多个后继。

✓ 结构特点：结点间有明显层次关系。

6.1 树的定义和二叉树

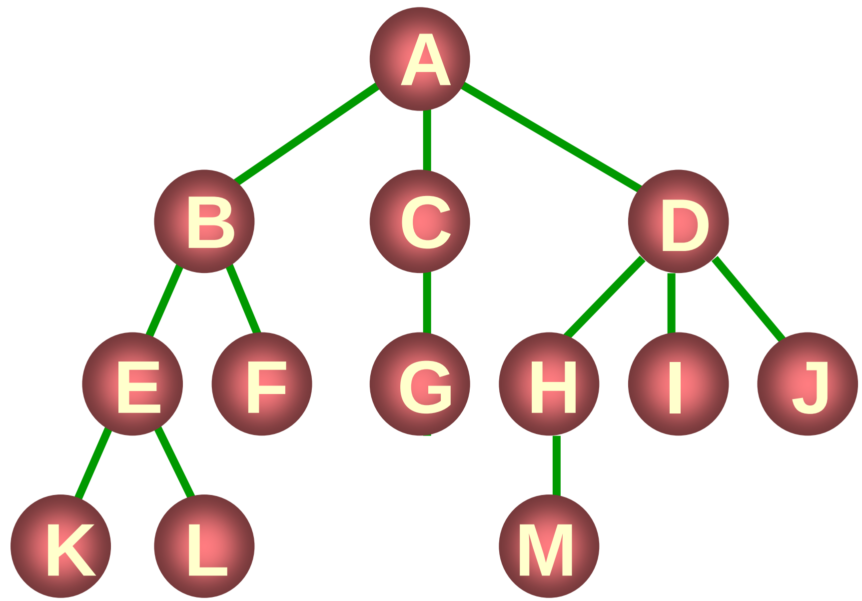
一. 树的定义

例如：A是根；其余结点分成三个互不相交的子集

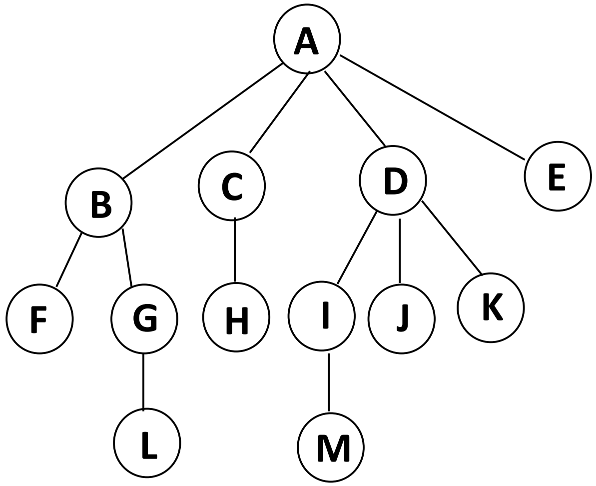
T1={B, E, F, K, L}； T2={C, G}； T3={D, H, I, J, M},

T1, T2, T3都是根A的子树，且本身也是一棵树。

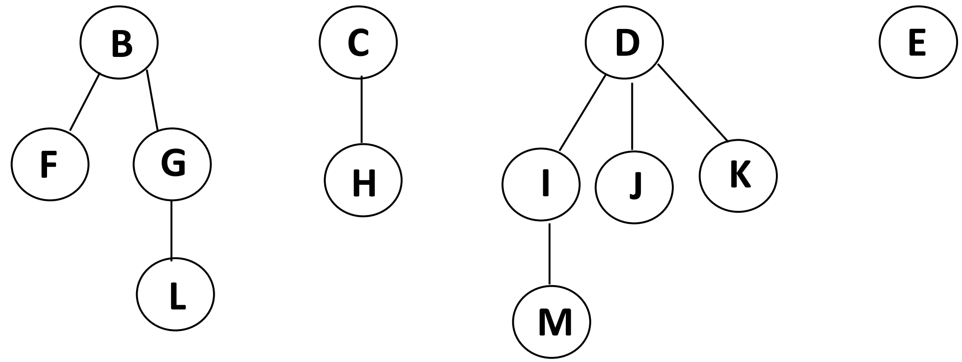
只有根结点的树



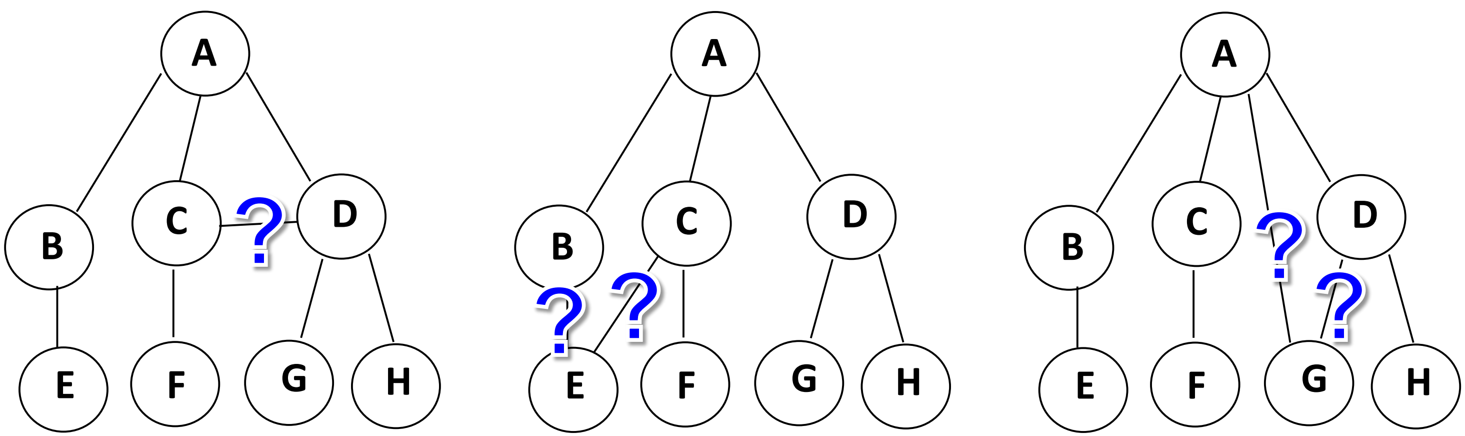
❖ 树与非树



(a) 树T



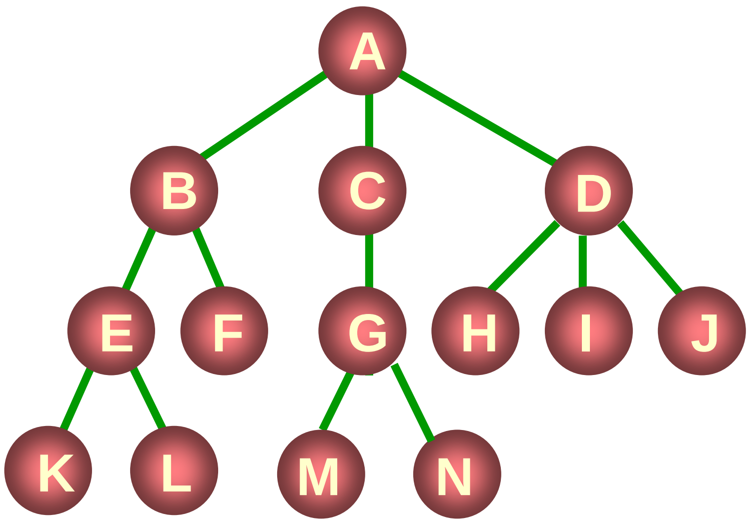
(b) 子树T_{A1}　(c) 子树T_{A2}　(d) 子树T_{A3}　(e)子树T_{A4}



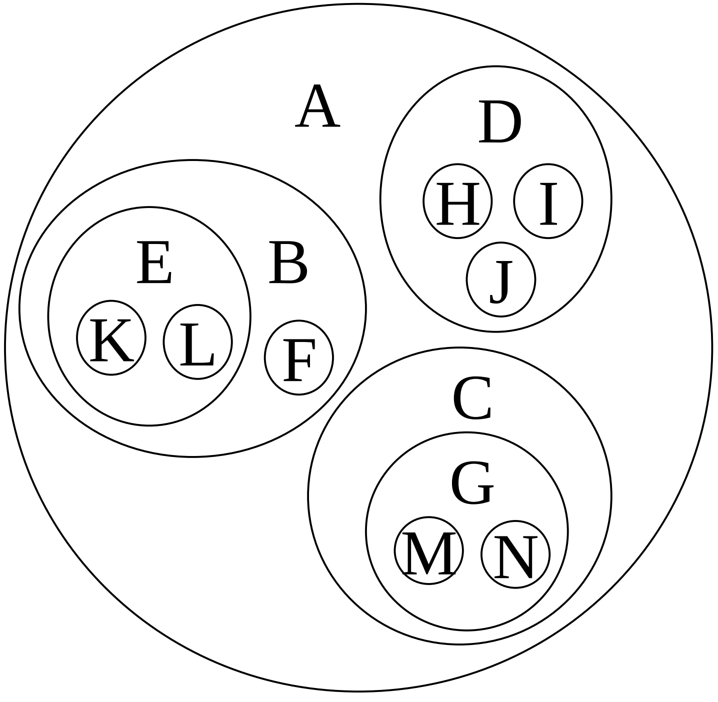
6.1 树的定义和二叉树

一. 树的定义

- 树的表示形式



(a) 树状图

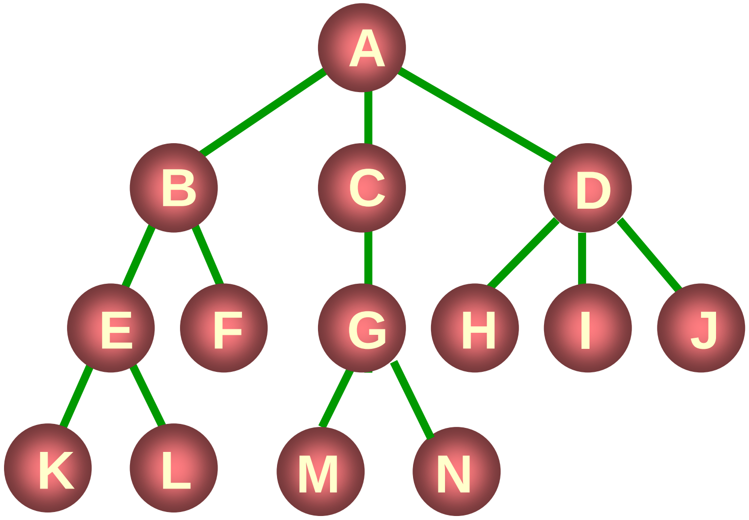


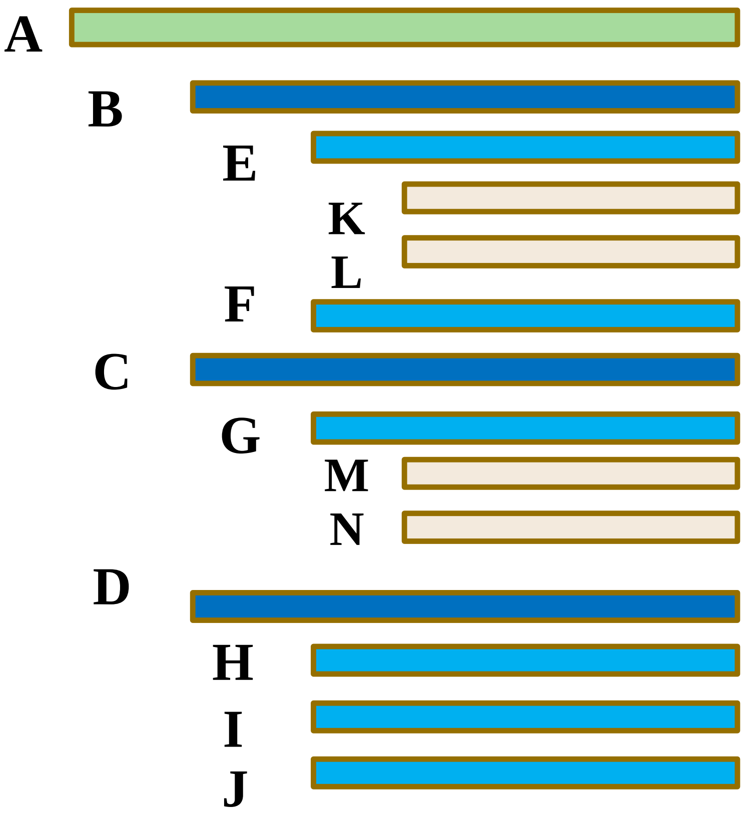
(b) 嵌套集合形式

6.1 树的定义和二叉树

一. 树的定义

- 树的表示形式





(c)凹入表示法

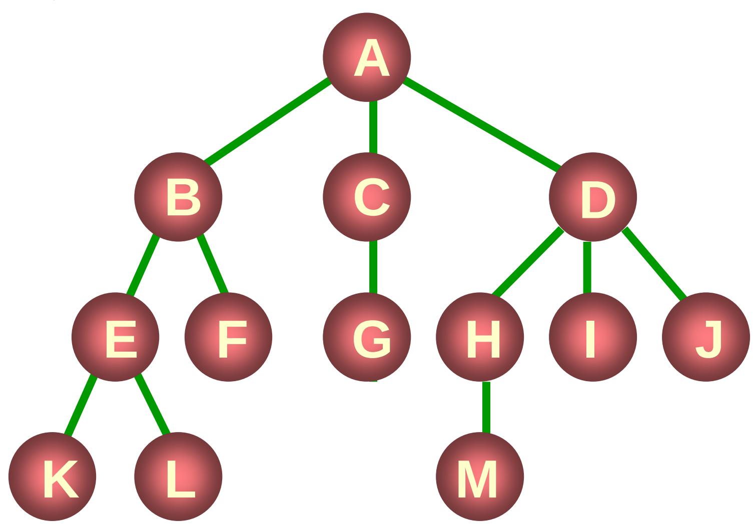
(A(B(E(K,L),F),C(G(M,N)),D(H,I,J))

(d) 广义表形式

6.1 树的定义和二叉树

二. 树的基本术语

- **结点：**包含一个数据元素及若干指向其子树的分支（结点之间的连接）。
- **结点的度：**一个结点所拥有的子树的个数。
- **树的度：**树中各结点的度的<u>最大值</u>，通常将度为m的树称为m次树或者m叉树。

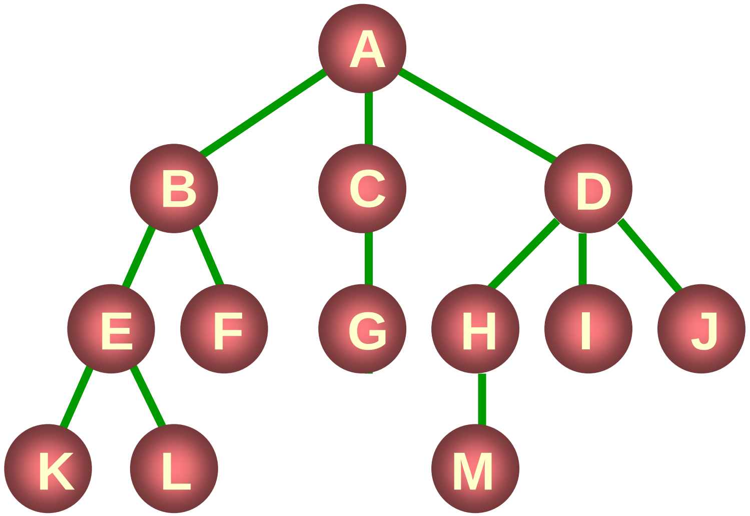


3次树

6.1 树的定义和二叉树

二. 树的基本术语

- **叶结点：**度为0的结点[没有子树的结点]，也称为终端结点。
- **分支结点：**度不为0的结点[包括根结点]，也称为非终端结点，除根外称为内部结点。

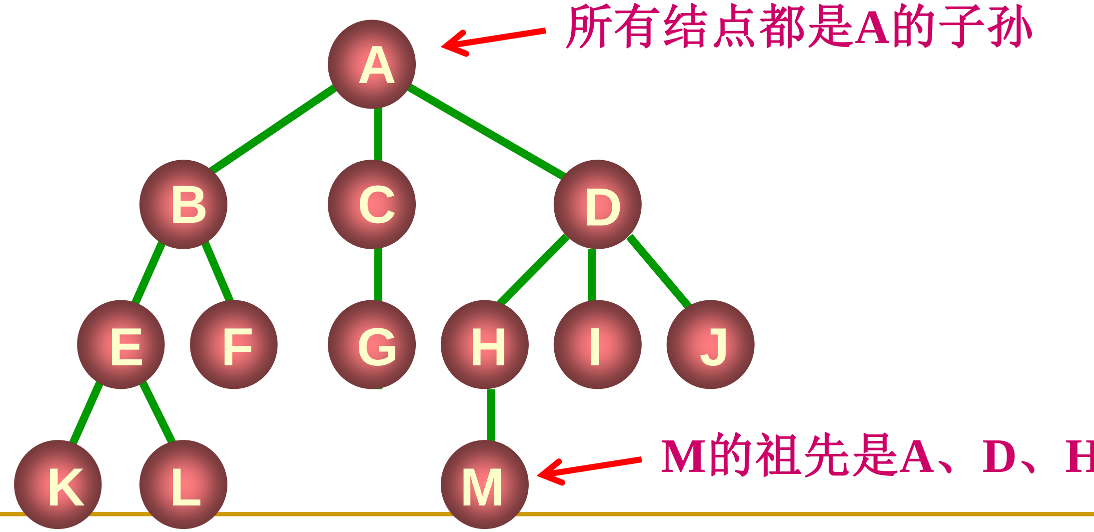


6.1 树的定义和二叉树

二．树的基本术语

- **孩子：** 结点的子树的根[直接后继，可能有多个]，而该结点称为孩子的双亲（孩子的直接前驱[最多只能有一个]）。
- **兄弟：** 同一双亲的孩子之间互称兄弟。
- **子孙：** 以某结点为根的子树中的任一结点称为该结点的子孙。
- **祖先：** 从根到该结点所经分支上的所有结点称为该结点的祖先。

A 的孩子结点有B 、C 、D
B 、C 、D 的双亲结点为A
B 、C 、D 互为兄弟结点

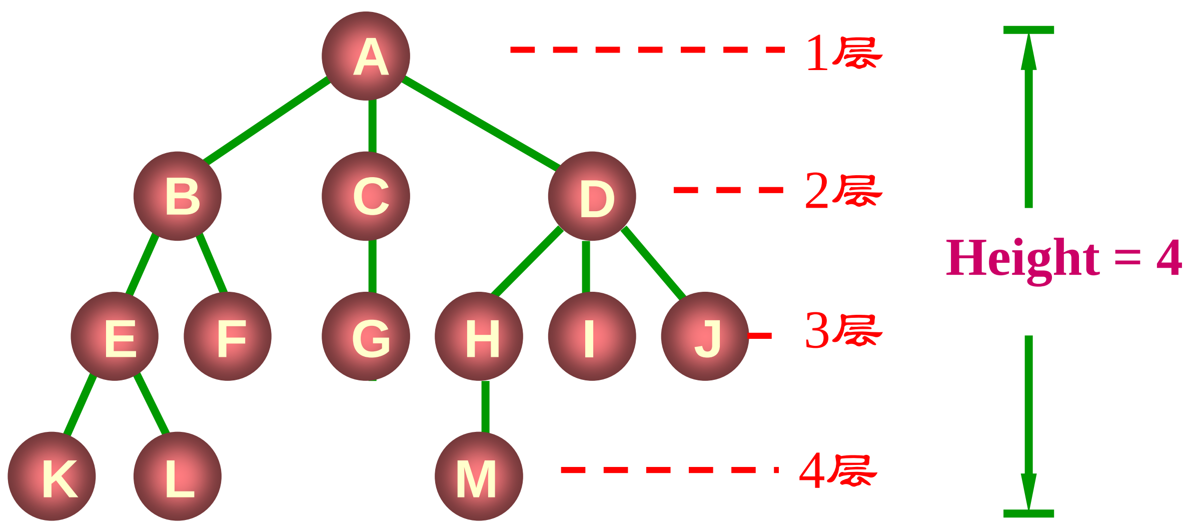


6.1 树的定义和二叉树

二. 树的基本术语

- **层次：**结点的层次从树根开始定义，根结点为第1层，其孩子为第2层，依此类推。

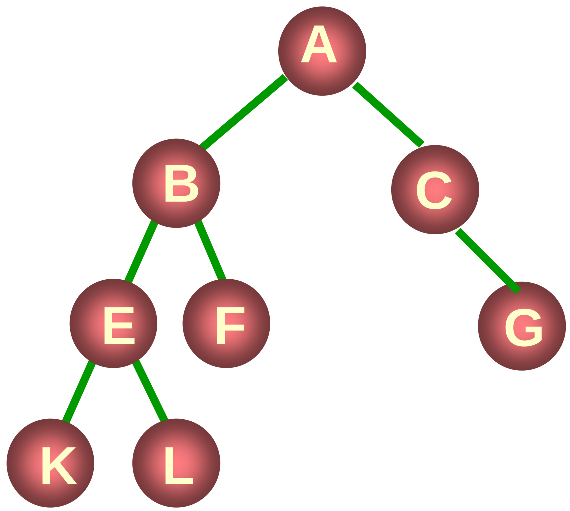
- **深度：**树中结点的<u>最大层次</u>称为树的深度（或树的高度）。

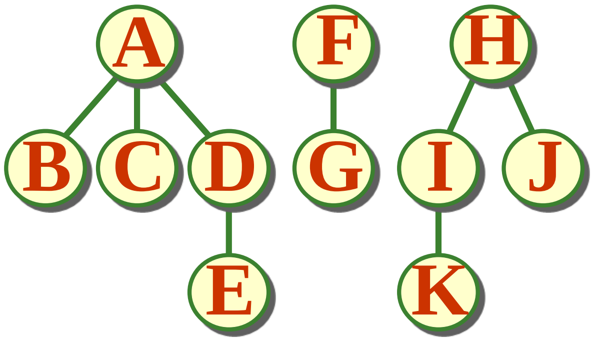


6.1 树的定义和二叉树

二. 树的基本术语

- **有序树：** 子树之间从左至右存在确定的次序关系（即不能互换）。
- **无序树：** 子树之间不存在确定的次序关系。
- **森林：** 互不相交的树的集合。对树中每个结点而言，其子树的集合即为森林。





6.1 树的定义和二叉树

三. 树的性质

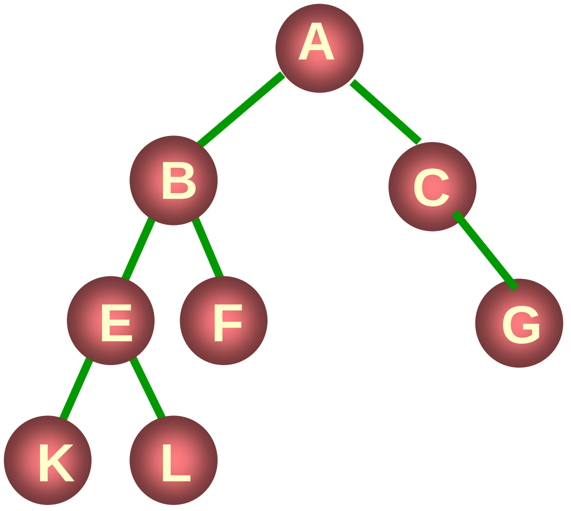
其一：树中的结点数为n，所有结点的度数之和为m，则m = n-1。

因为：

（1）所有结点的度之和 m ＝ 分支数B

（2）由于除根节点以外的每个结点，都对应一个与其双亲结点相联系的分支，所以分支数B+1与结点数n相同，即 B+1 = n,

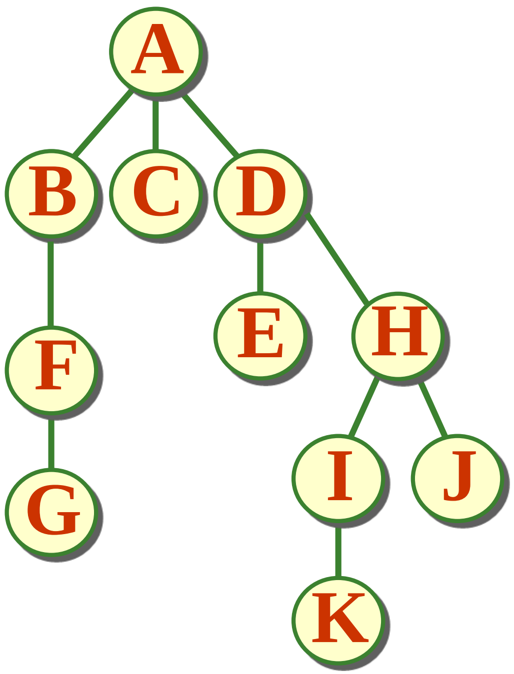
所以 m = n-1



6.1 树的定义和二叉树

练习：

- 已知一颗树如图所示，求
  - 树的度和深度
  - 叶子结点和分支结点的数量
  - 结点H的度
  - 结点D的子孙
  - 结点I的祖先



6.2 二叉树

一. 二叉树的定义

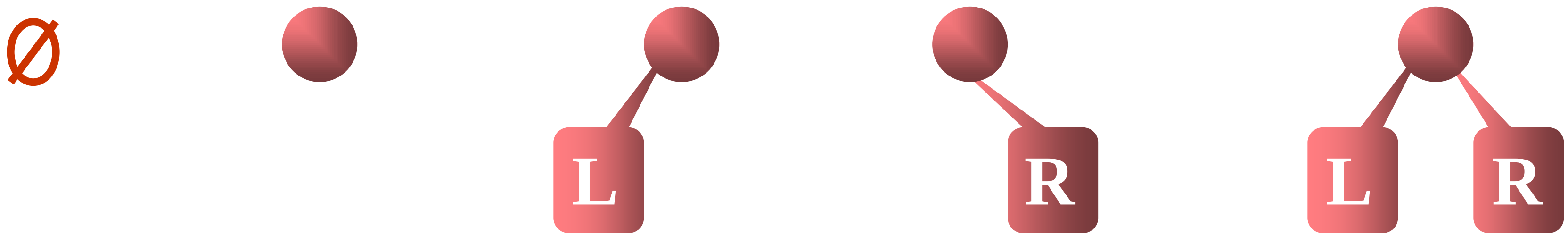
- 二叉树是一种特殊的树，一棵二叉树是结点的一个有限集合，该集合或者为空，或者是由一个根结点加上两棵分别称为左子树和右子树的二叉树组成。
- 特点：
  - 每个结点的度≤2；
  - 二叉树是有序树，每个结点的子树有左右之分。

研究二叉树的意义？

问题转化：将树转换为二叉树，从而利用二叉树解决树的有关问题。

6.2 二叉树

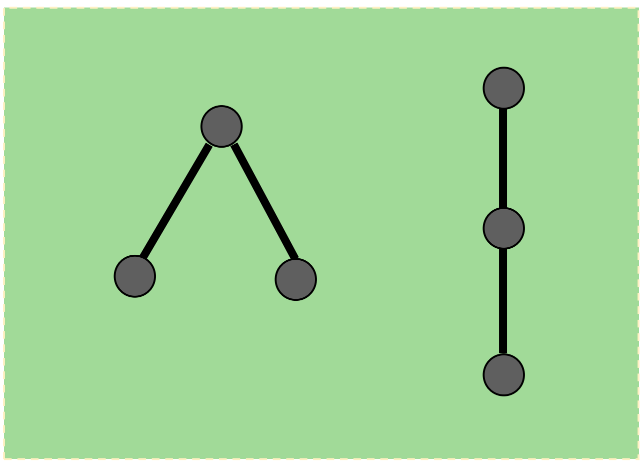
二叉树的五种不同形态



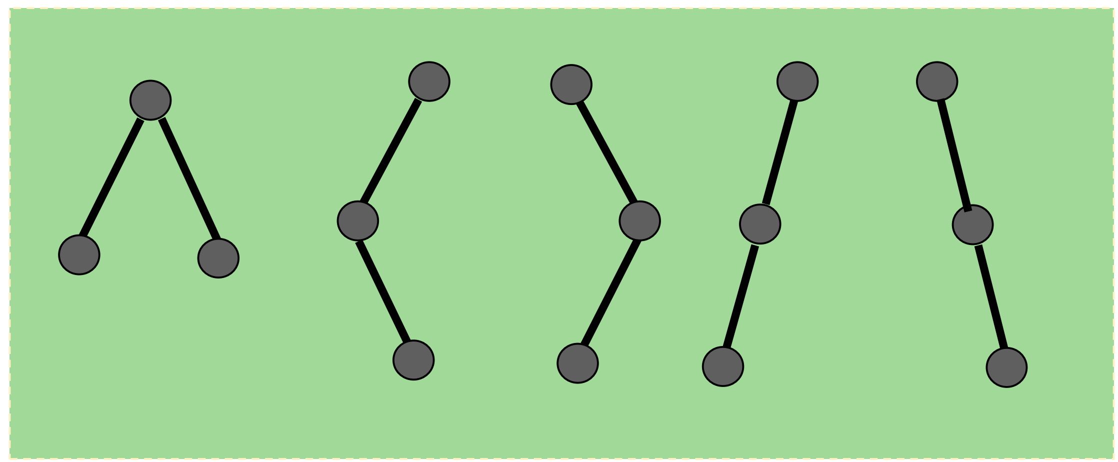
空树　　只有根　　只有左子树　　只有右子树　　有左右子树

6.2 二叉树

例：具有3个结点的树和具有3个结点的二叉树的形态。



3个结点的树



3个结点的二叉树

6.2 二叉树

二． 二叉树的性质

性质1 在二叉树的第i层上至多有2^{i-1}个结点。

- 证明： **[用数学归纳法]**

1. i=1，只有一个根结点，因此$2^{i-1}=2^0=1$
2. 设第i-1层上，以上性质成立，即第i-1层至多有$2^{(i-1)-1}$结点。由二叉树的定义可知，任何结点的度不大于2，因此，第i层上的结点数最多为第i-1层上的两倍，即$2*2^{i-2}=2^{i-1}$

6.2 二叉树

二．二叉树的性质

性质2 深度为k的二叉树至多有2^k-1个结点

证明： **[用求等比级数前*k*项和的公式]**

1. 由性质 1，已知第i层上结点数最多为2^{i-1}

2. 那么，深度为k的二叉树的结点总数至多有

$$\sum_{i=1}^{k} 2^{i-1} = 2^k-1$$

二． 二叉树的性质

性质3 如果二叉树终端结点数为n_0，度为2的结点数为n_2，

则 $n_0=n_2+1$。

证明：

1. 设度为1的结点个数是n_1，则总结点数$n= n_0+n_1+n_2$
2. 设B为二叉树的分支数，则总结点数$n=B+1$
3. 每个分支皆由度为1或2的结点发出，即 $B = n_1+2n_2$

则有： $n = B+1 = n_1+2n_2 + 1$

即 $n_0 +n_1+n_2 = n_1+2n_2+1$

因此 $n_0 = n_2+1$

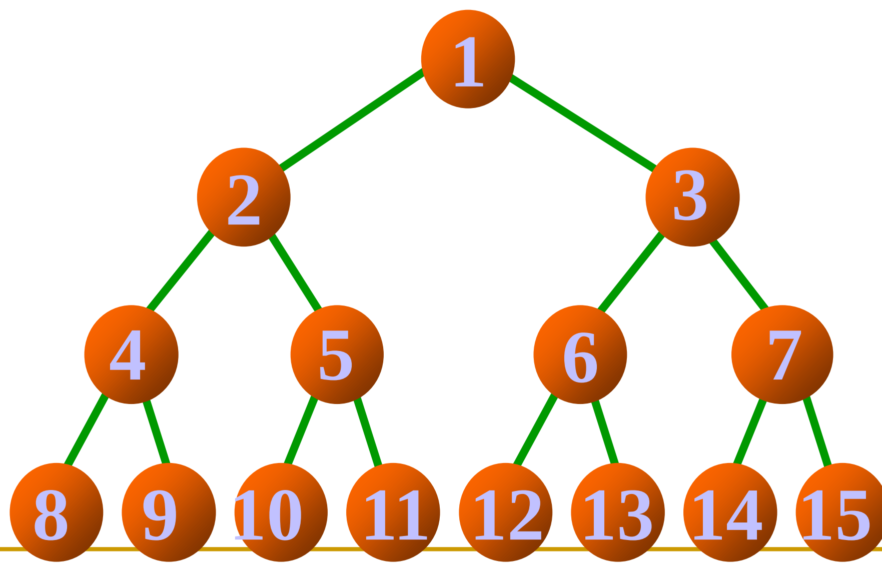
6.2 二叉树

三. 满二叉树

- 定义：一个深度为k且有2^k-1个结点的二叉树。

  特点：每层上的结点数都是最大数2^{k-1}

- 编号：从根结点起，自上而下、自左至右的次序，依次给结点编以从1开始的连续自然数。

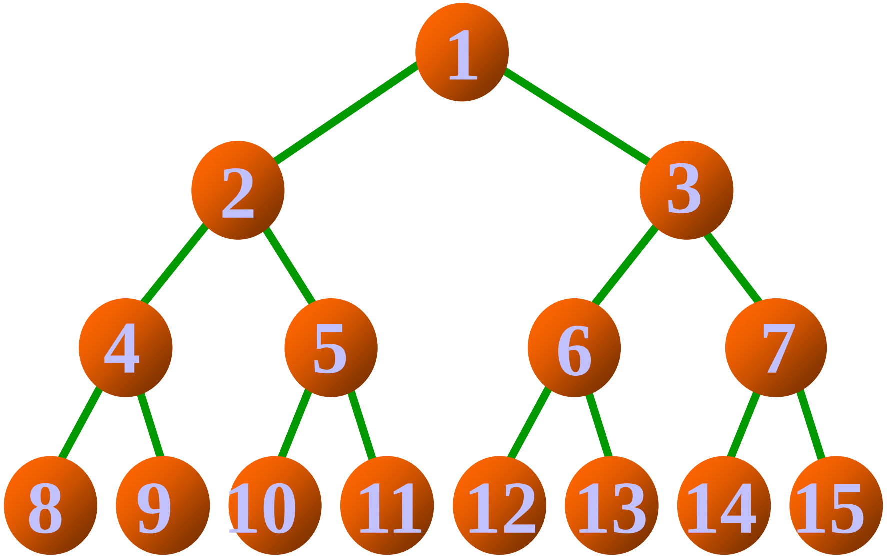


6.2 二叉树

三. 满二叉树

特点：

1. 只有度为0和度为2的结点，所有分支结点都有左右子树;
2. 每层上的结点数都达到最大值2^{k-1}
3. 叶子只能出现在最下一层。

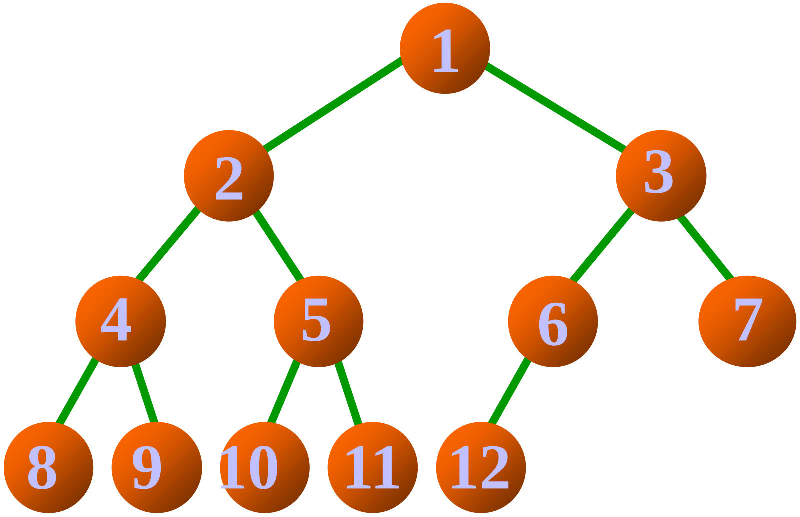


编号：从根结点起，自上而下、自左至右的次序，依次给结点编以从1开始的连续自然数。

6.2 二叉树

四. 完全二叉树

- 定义：深度为k且有n个结点的二叉树，当且仅当每一个结点都与深度相同的满二叉树中编号从1到n的结点一一对应。
- 特点：
  - 叶子结点只可能在最大两层上出现
  - 左子树的深度或等于右子树的深度，或者比右子树的深度多 1 。



6.2 二叉树

四. 完全二叉树

性质4 具有n个结点的完全二叉树，其深度为$\lfloor \log_2 n \rfloor+1$。

证明：

设完全二叉树的深度为k，

由二叉树性质 2， 则有

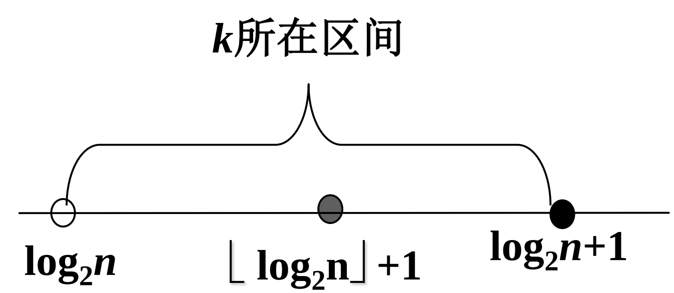
$2^{k-1}-1 < n \leqslant 2^k-1$

取对数 $k-1 \leqslant \log_2 n < k$

即：

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

由于k是整数，所以必有 $k = \lfloor \log_2 n \rfloor + 1$



6.2 二叉树

四. 完全二叉树

性质5 如果将一棵有n个结点的完全二叉树的结点按层序(自顶向下，同一层自左向右）连续编号1, 2, …, n， 则对任一结点i ($1 \le i \le n$)，有以下关系：

- 若$i = 1$, 则 i 是二叉树的根，无双亲
- 若$i > 1$, 则 i 的双亲结点为$\lfloor i/2 \rfloor$
- 若$2*i \le n$, 则 i 的左孩子为$2*i$，否则无左孩子

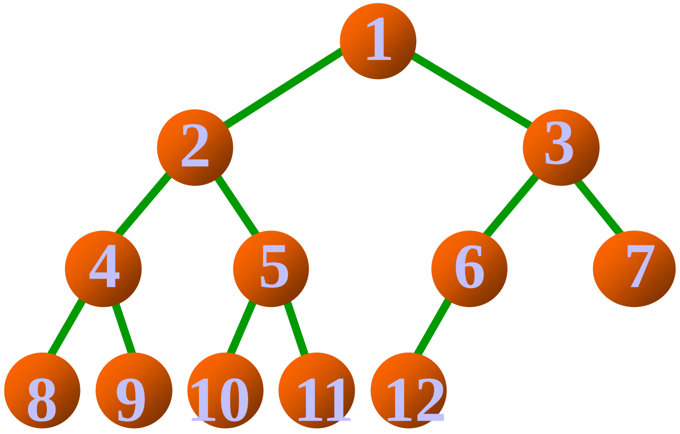
  若$2*i+1 \le n$, 则 i 的右孩子为$2*i+1$，否则无右孩子
- i 所在层次为$\lfloor \log_2 i \rfloor +1$

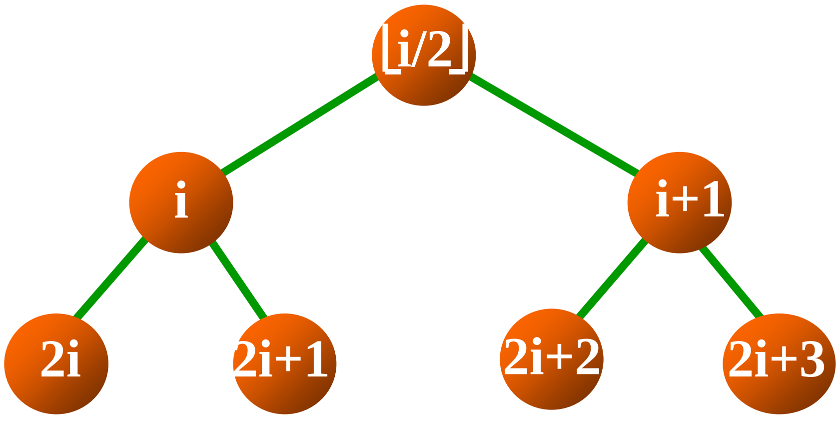
6.2 二叉树

四. 完全二叉树

性质5 在完全二叉树中，

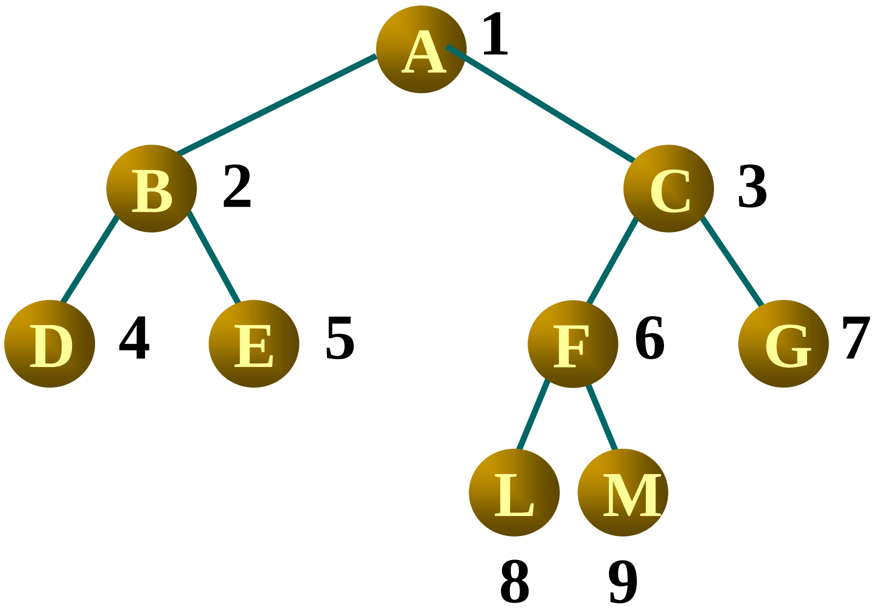
- 结点i的双亲为 $\lfloor i/2 \rfloor$
- 结点i的左孩子LCHILD(i)=2i
- 结点i的右孩子RCHILD(i)=2i+1





6.2 二叉树

练习：下图所示二叉树是满二叉树吗？



满二叉树在同样深度的二叉树中结点个数最多
满二叉树在同样深度的二叉树中叶子结点个数最多

6.2 二叉树

练习：

1. 已知一颗完全二叉树第7层有20个结点，则整棵树的结点数?

2. 已知一棵完全二叉树有100个结点，根节点编号为1，按层次遍历编号，则结点45的父亲编号为？结点50的孩子编号情况如何?

3. 设深度为h的二叉树上只有叶子结点和同时具有左右子树的结点，则此类二叉树中所包含的结点数目至少为______

   A. 2h　　　　B. 2h-2

   C. 2h＋1　　　D. 2h-1

6.2 二叉树

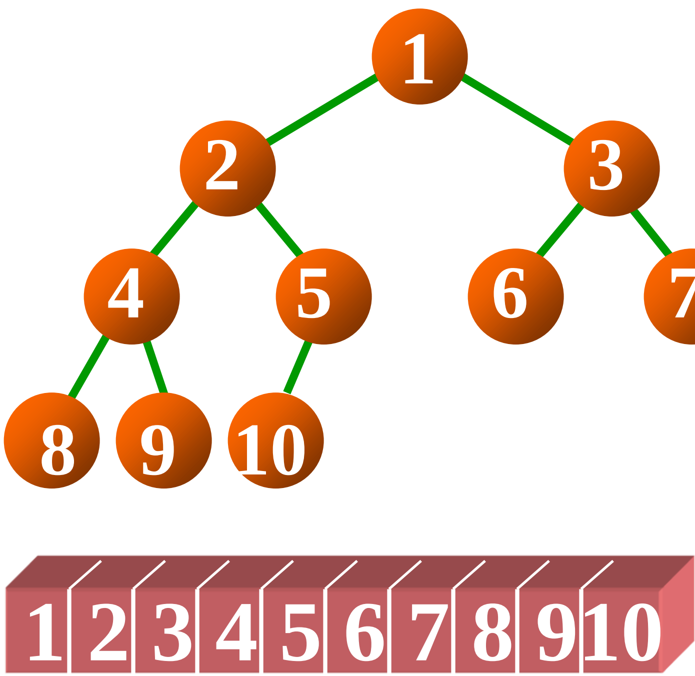
五. 二叉树的顺序存储结构

- 用一组连续的存储单元依次自上而下,自左至右存储结点

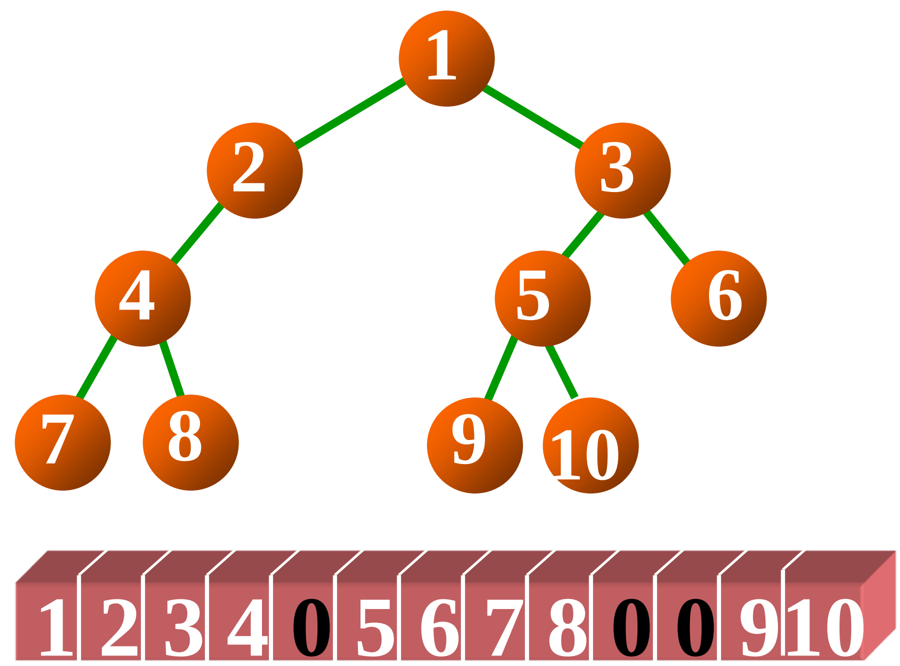
```
#define MAX_TREE_SIZE 100 //最大结点数
Typedef TElemType SqBiTree[MAX_TREE_SIZE];
SqBiTree bt;
```

6.2 二叉树

五. 二叉树的顺序存储结构



完全二叉树的顺序表示



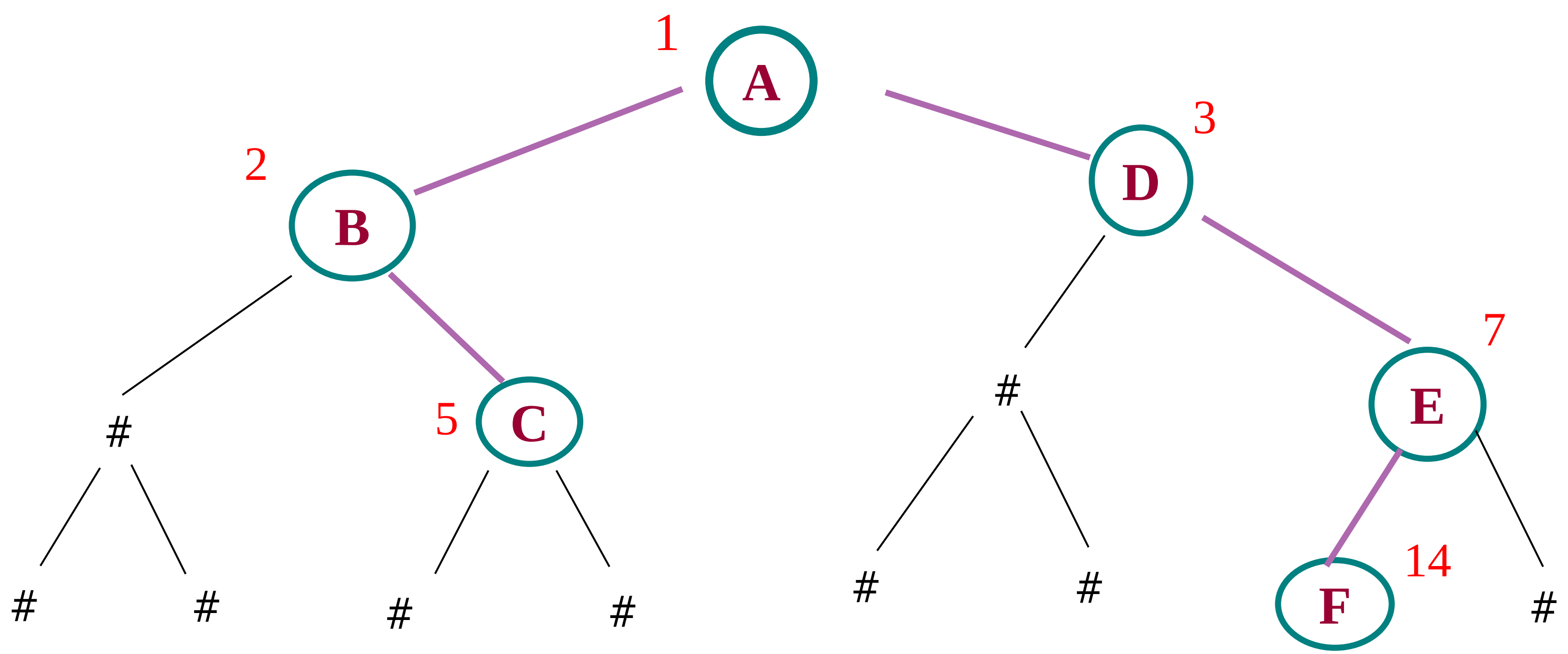
一般二叉树的顺序表示

6.2 二叉树

五. 二叉树的顺序存储结构

- 用一组连续的存储单元依次自上而下, 自左至右存储结点

例:



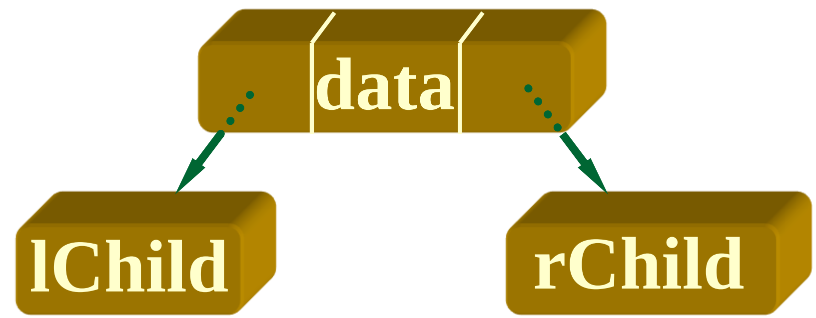
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	D	#	C	#	E	#	#	#	#	#	#	F	#

二叉树的顺序存储缺点：浪费空间。

6.2 二叉树

六. 二叉树的链式存储结构

- 采用二叉链表，数据域加上左、右孩子指针



6.2 二叉树

六. 二叉树的链式存储结构

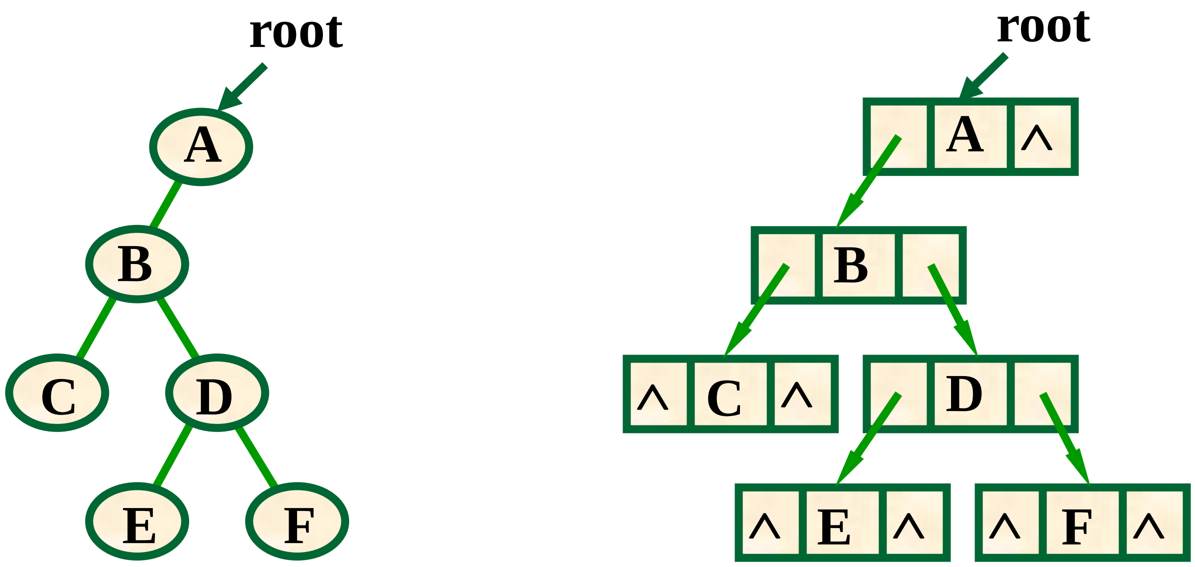
- 二叉链表由一个个结点组成，二叉链表结点由一个数据域和两个指针域组成

```
typedef struct BiTNode {
       TElemType        data;
       struct BiTNode  *lChild, *rChild;
} BiTNode, * BiTree
```

6.2 二叉树

六. 二叉树的链式存储结构

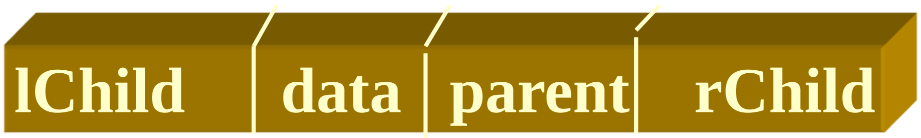
- 二叉链表以及存储表示

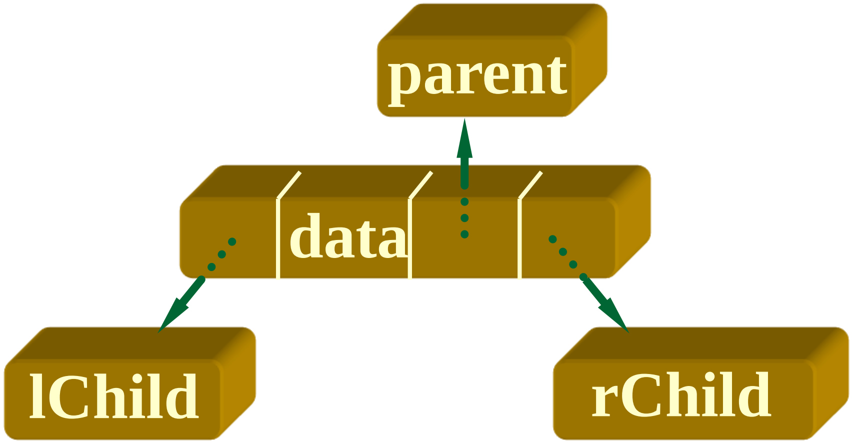


6.2 二叉树

六. 二叉树的链式存储结构

- 三叉链表, 采用数据域加上左、右孩子指针及双亲指针

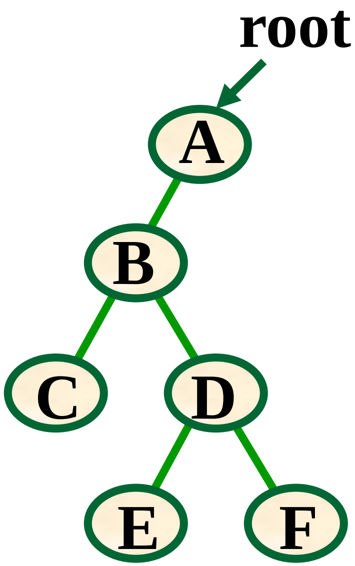


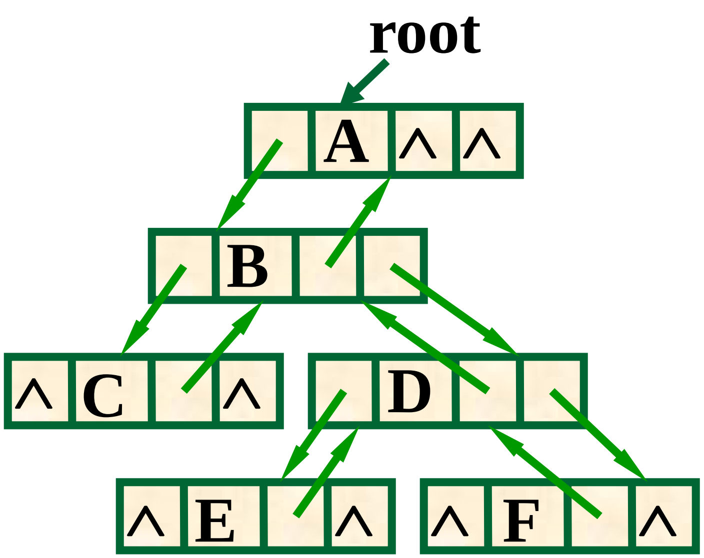


6.2 二叉树

六. 二叉树的链式存储结构

- 三叉链表以及存储表示





6.2 二叉树

七. 树的基本运算

- 创建树： 根据树的表示方法字符串生成树的存储结构
- 查找：满足某种特定关系的结点，如查找当前结点的双亲结点等；
- 插入或删除某个结点：如在树的当前结点上插入一个新结点或删除当前结点的第i个孩子结点等；
- 遍历

练习

1. 在二叉链表中，指针p指向的结点是叶子，则p满足条件？

   p->Lchild=Null p->Rchild=Null

2. 在含有n个结点的二叉链表中，有多少个空指针域?

   有n+1个空指针域