第六章 树和二叉树

制作：数据结构在线课程课题组

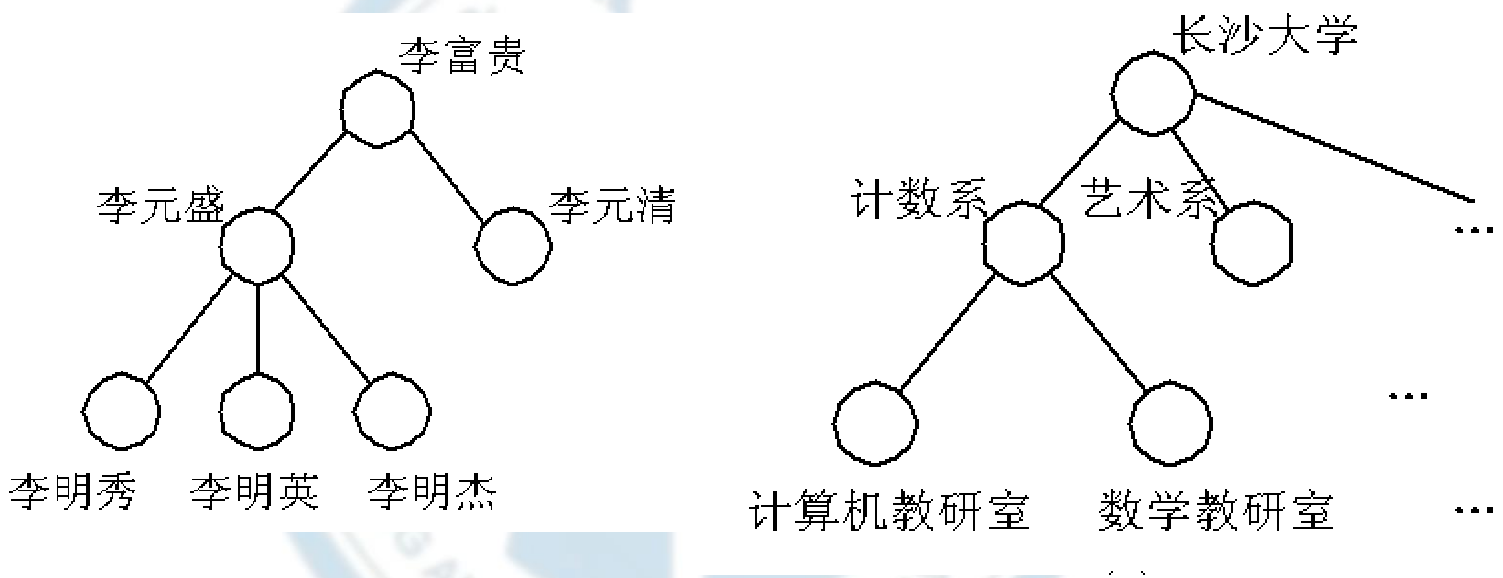
南京审计大学 信息工程学院

2020. 10

- 领会树和二叉树的类型定义，理解树和二叉树的结构差别。
- 熟记二叉树的主要特性，并掌握它们的证明方法。
- 熟练掌握二叉树的各种遍历算法，并能灵活运用遍历算法实现二叉树的其它操作。
- 熟练掌握二叉树和树的二叉链表存储结构及其建立的算法。
- 学会编写实现树的各种操作的算法。
- 了解最优树的特性，掌握建立最优树和赫夫曼编码的方法。

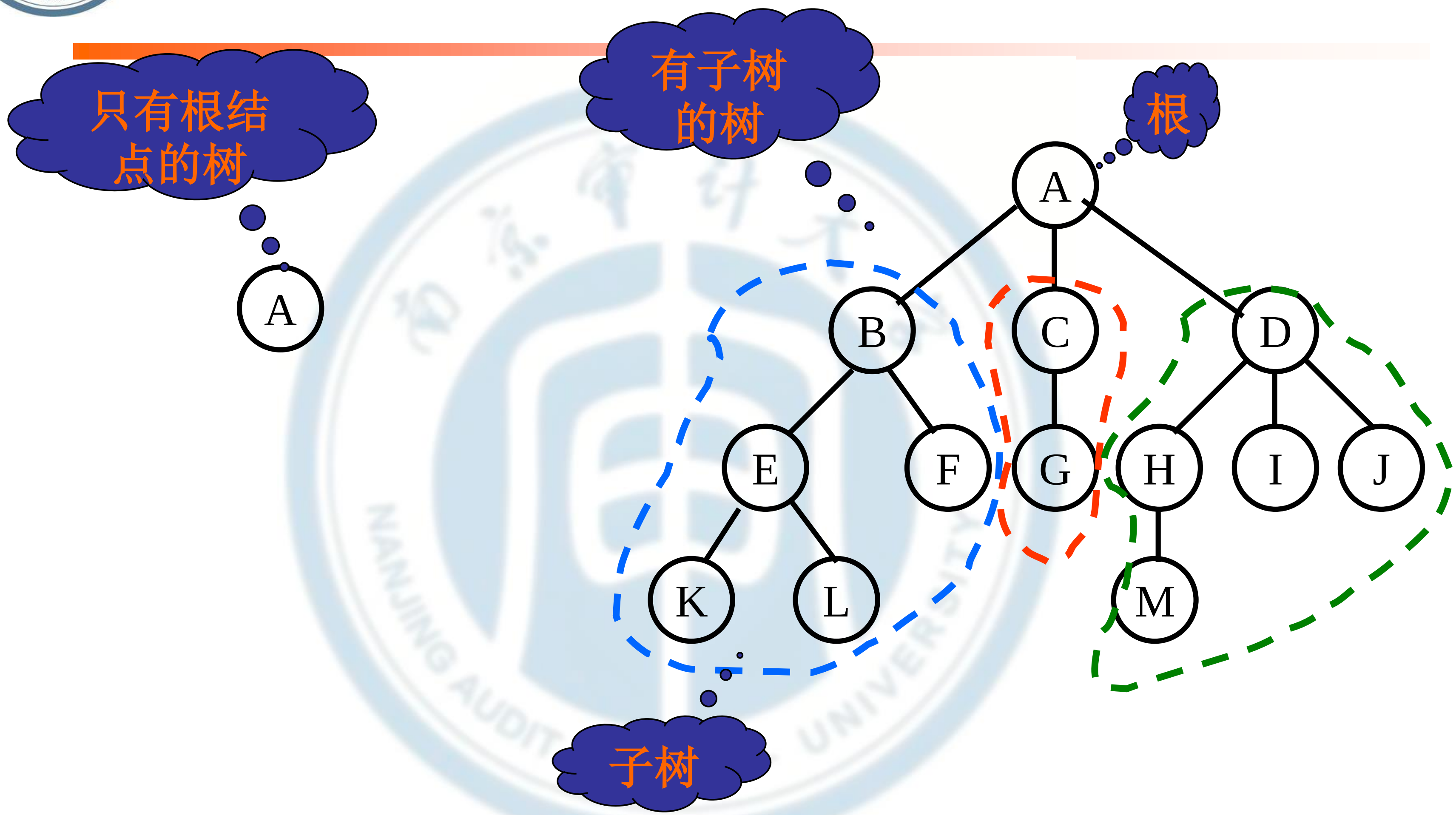
认识树型结构

树型结构是一类非常重要的非线性数据结构。



6.1 树的定义和基本术语

- 树(tree)
  - 是n(n≥0)个结点的有限集T；
  - 在任意一棵非空树中，
    - 有且仅有一个特定的结点，称为树的根(root)；
    - 当n>1时，其余结点可分为m(m>0)个互不相交的有限集T1,T2,……Tm，其中每一个集合本身又是一棵树，称为根的子树(subtree)。
  - 特点：
    - 非空树中至少有一个结点——根；
    - 树中各子树是互不相交的集合。


只有根结点的树
A
有子树的树
根
A
B
C
D
E
F
G
H
I
J
K
L
M
子树

树的抽象数据类型的定义

ADT Tree {

数据对象：D是具有相同特性的数据元素的集合。

数据关系：

若 D 为空集，则称为空树；

若 D 中仅含一个数据元素，则关系R为空集；

若 D 中含多于一个数据元素，则 R={H}，H是如下二元关系：

(1) 在D中存在唯一的称为根的数据元素 root，它在关系H下无前驱；

(2) 当n>1时，其余数据元素可分为 m(m>0) 个互不相交的(非空)有限集 T1,T2,...,Tm，其中每一个子集本身又是一棵符合本定义的树，称为根 root 的子树，每一棵子树的根xi都是根root的后继，即 <root,xi>∈H，i=1,2,...,m。

基本术语

- 结点(node)：包含一个数据元素及若干指向其子树的分支。
- 结点的度(degree)：结点拥有的子树数。
- 叶子(leaf)：度为0的结点。
- 分支结点：度不为0的结点。
- 树的度：一棵树中各结点的度的最大值。
- 孩子(child)：结点的子树的根称为该结点的孩子。
- 双亲(parents)：孩子结点的上层结点。
- 兄弟(sibling)：同一双亲的孩子之间互称为兄弟。
- 祖先：从根结点到该结点所经分支上的所有结点。
- 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。

将树中结点的各子树看成从左至右是有次序的，即不能互换

- 结点的层次(level)：从根结点算起，根为第一层，根的孩子为第二层……。
- 堂兄弟：其双亲在同一层的结点互为堂兄弟。
- 深度(depth)：树中结点的最大层次。
- 有序树：将树中结点的各子树看成从左至右是有次序的，即不能互换。
- 无序树：将树中结点的各子树看成从左至右是无次序的，即可以互换。
- 森林(forest)： m(m≥0)棵互不相交的树的集合。

森林：

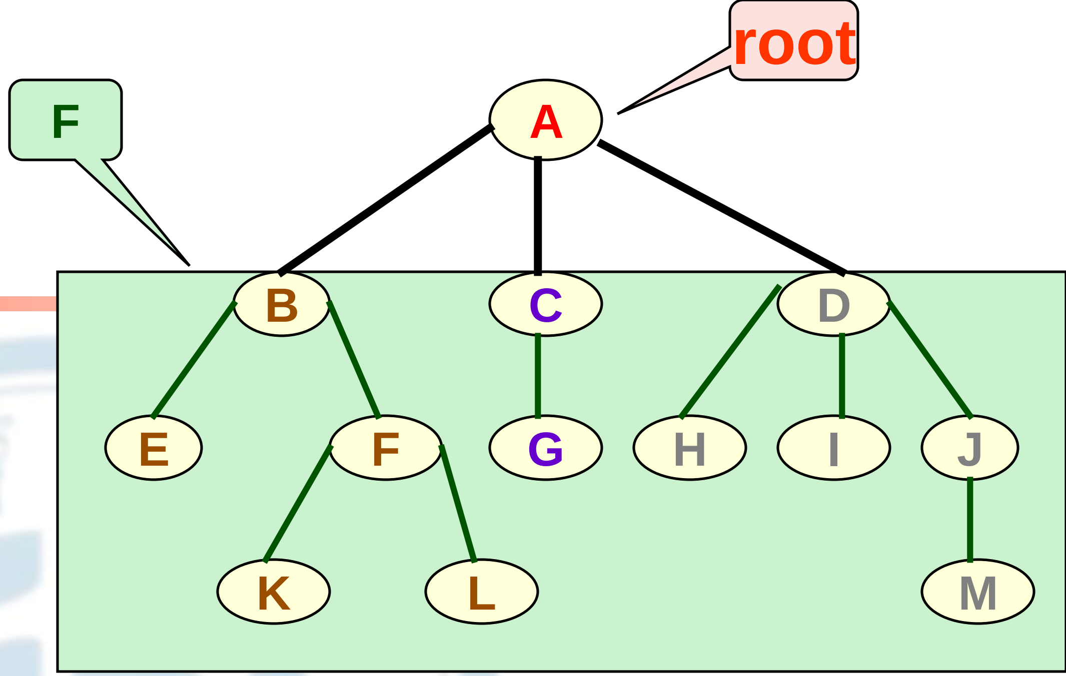
是m（m≥0）棵互不相交的树的集合。

任何一棵非空树是一个二元组

Tree = （root，F）

其中：root 被称为根结点，

F 被称为子树森林。



结点A的度：3
结点B的度：2
结点M的度：0

树的度：3

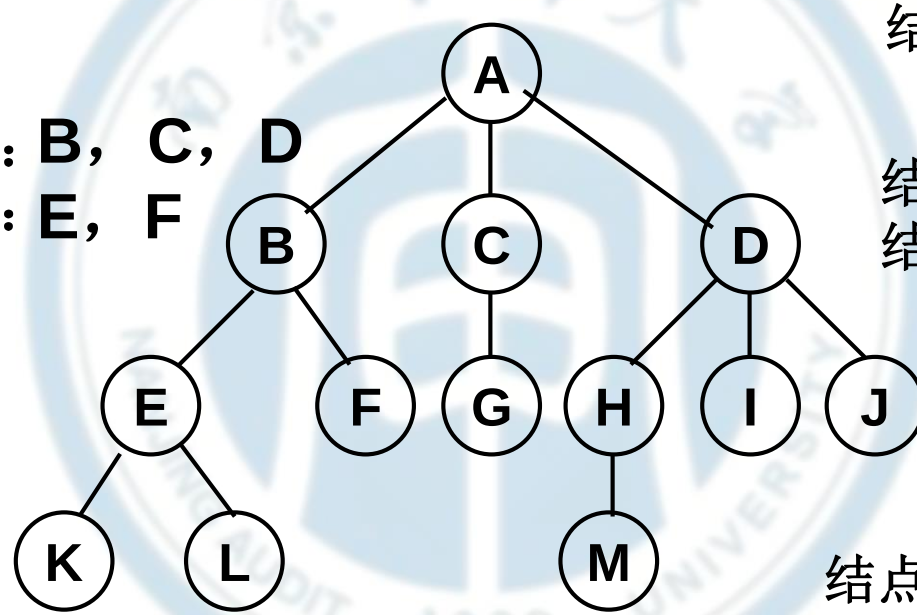
结点A的孩子：B，C，D
结点B的孩子：E，F

结点A的层次：1
结点M的层次：4

叶子：K，L，F，G，M，I，J

结点I的双亲：D
结点L的双亲：E

结点B，C，D为兄弟
结点K，L为兄弟

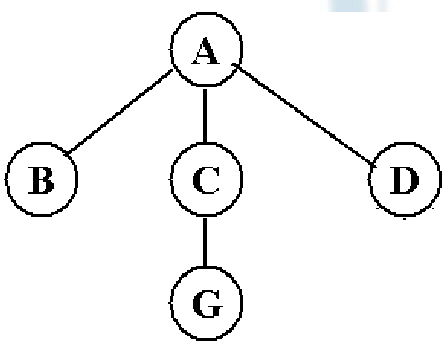


结点A是 结点F，G 的祖先

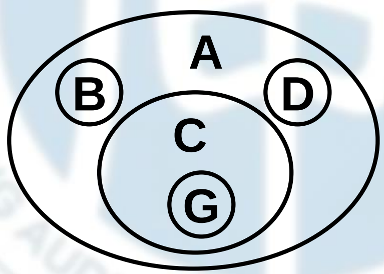
结点F，G为堂兄弟

树的深度：4

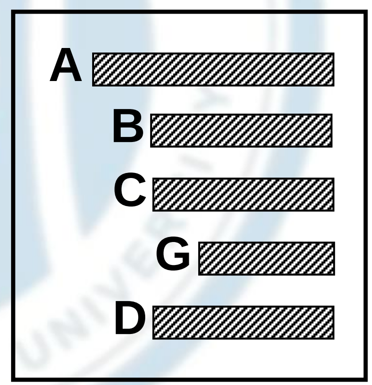
- 树的表示方法
  - 树形表示法：自然界倒长的树；
  - 文氏表示法：用集合表示；
  - 凹入表示法：类似书目；
  - 嵌套括号表示法：广义表表示法。



树形



文氏



凹入

(A(B,C(G),D))

嵌套括号

基本操作：

InitTree(&T);

操作结果：构造空树 T。

CreateTree(&T,definition);

初始条件：definition 给出树T的定义。

操作结果：按 definition 构造树 T。

DestroyTree(&T);

初始条件：树 T 存在。

操作结果：销毁树 T。

TreeEmpty(T);

初始条件：树 T 存在。

操作结果：若 T 为空树，则返回 TRUE，否则返回 FALSE。

TreeDepth(T);

初始条件：树T存在。

操作结果：返回T的深度。

Root(T);

初始条件：树 T 存在。

操作结果：返回 T 的根。

Value(T, cur_e);

初始条件：树 T 存在，cur_e 是 T 中某个结点。

操作结果：返回 cur_e 的值。

Parent(T, cur_e);

初始条件：树 T 存在，cur_e 是 T 中某个结点。

操作结果：若 cur_e 是T的非根结点，则返回它的双亲，否则返回“空”。

LeftChild(T, cur_e);

初始条件：树 T 存在，cur_e 是 T 中某个结点。

操作结果：若 cur_e 是T的非叶子结点，则返回它的最左孩子，否则返回“空”。

RightSibling(T, cur_e);

初始条件：树 T 存在，cur_e 是 T 中某个结点。

操作结果：若 cur_e 有右兄弟，则返回它的右兄弟，否则返回“空”。

TraverseTree(T, visit());

初始条件：树T存在，visit 是对结点操作的应用函数。

操作结果：按某种次序对 T 的每个结点调用函数 visit() 一次且至多一次。一旦 visit() 失败，则操作失败。

Assign(T, cur_e, value);

初始条件：树T存在，cur_e 是 T 中某个结点。

操作结果：结点 cur_e 赋值为 value。

ClearTree(&T);

初始条件：树 T 存在。

操作结果：将树 T 清为空树。

InsertChild(&T, &p, i, c);

初始条件：树 T 存在，p 指向T中某个结点，1≤i≤p 所指结点的度+1，非空树 c 与 T 不相交。

操作结果：插入 c 为 T 中 p 所指结点的第 i 棵子树。

DeleteChild(&T, &p, i);

初始条件：树 T 存在，p 指向 T 中某个结点，1≤i≤p 指结点的度。

操作结果：删除 T 中 p 所指结点的第 i 棵子树。

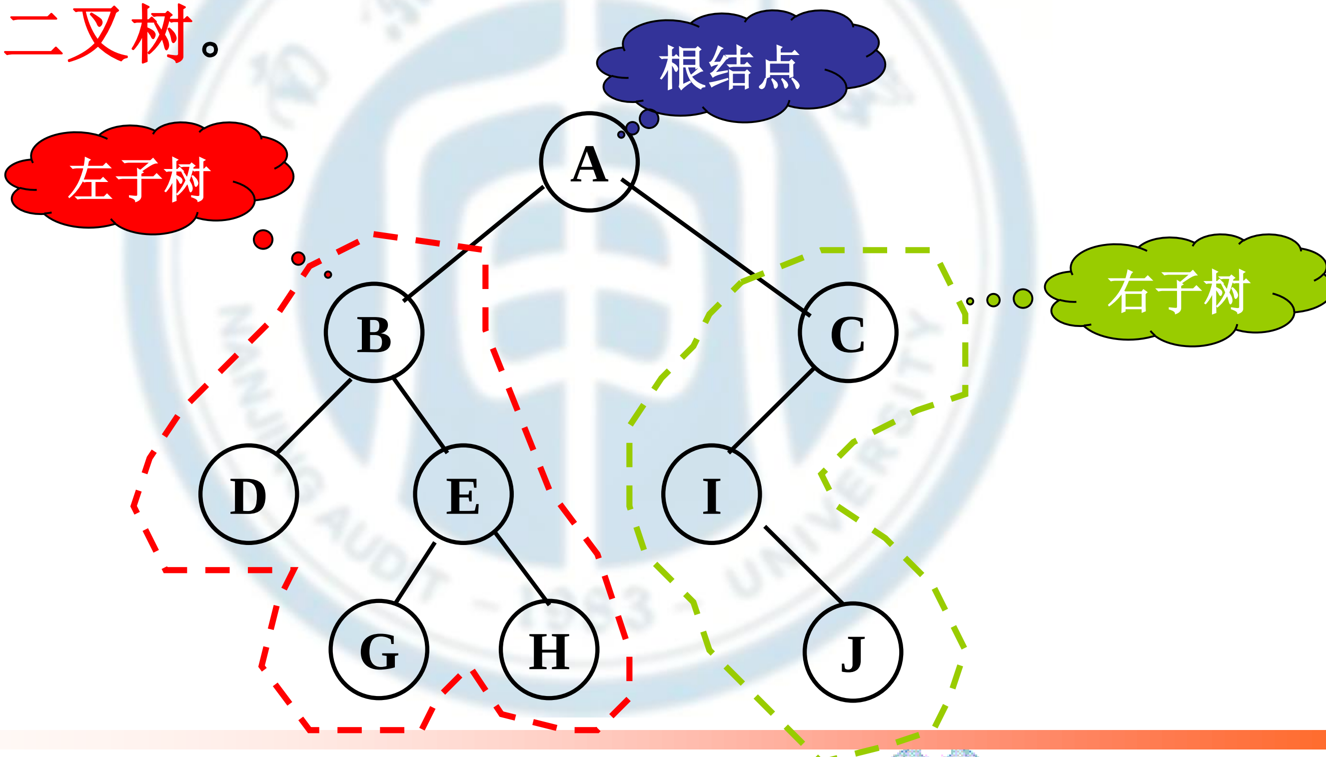
} ADT Tree

树和线性结构对照

线性结构	树结构
存在唯一的没有前驱的"首元素"	存在唯一的没有前驱的"根结点"
存在唯一的没有后继的"尾元素"	存在多个没有后继的"叶子"
其余元素均存在唯一的"前驱元素"和唯一的"后继元素"	其余结点均存在唯一的"前驱(双亲)结点"和多个"后继(孩子)结点"

6.2 二叉树

- 二叉树是n(n>=0)个结点的有限集，其子树分为互不相交的两个集合，分别称为左子树和右子树，左子树和右子树也是二叉树。



思考：二叉树与树的区别？

二叉树与无序树的区别

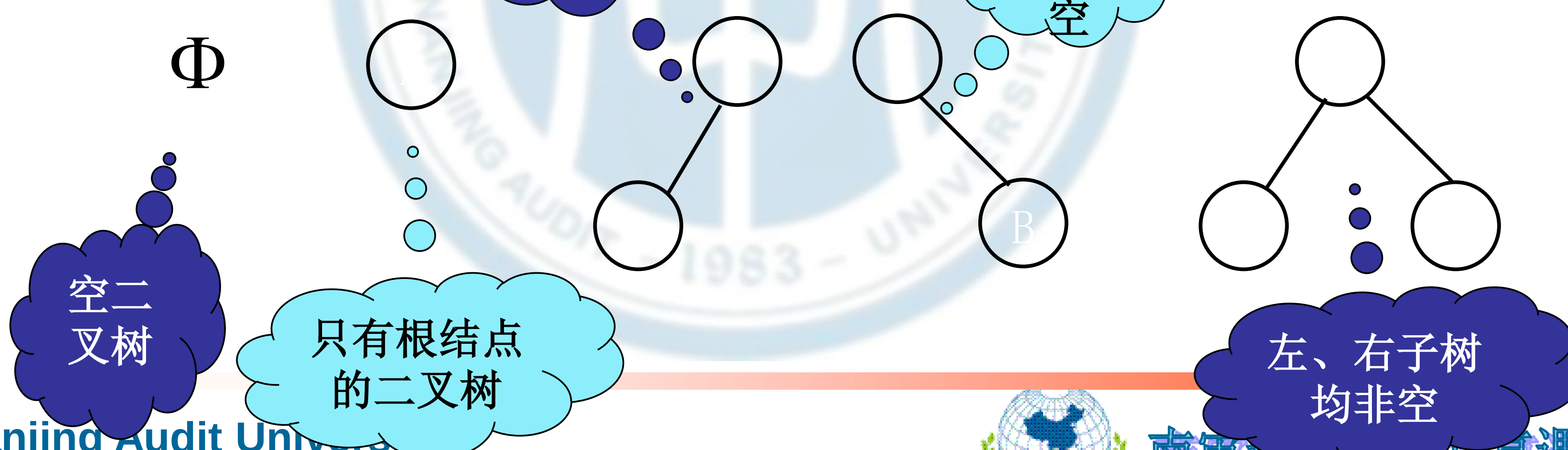
二叉树与有序树的区别

二叉树就是度为2的有序树吗？ 否

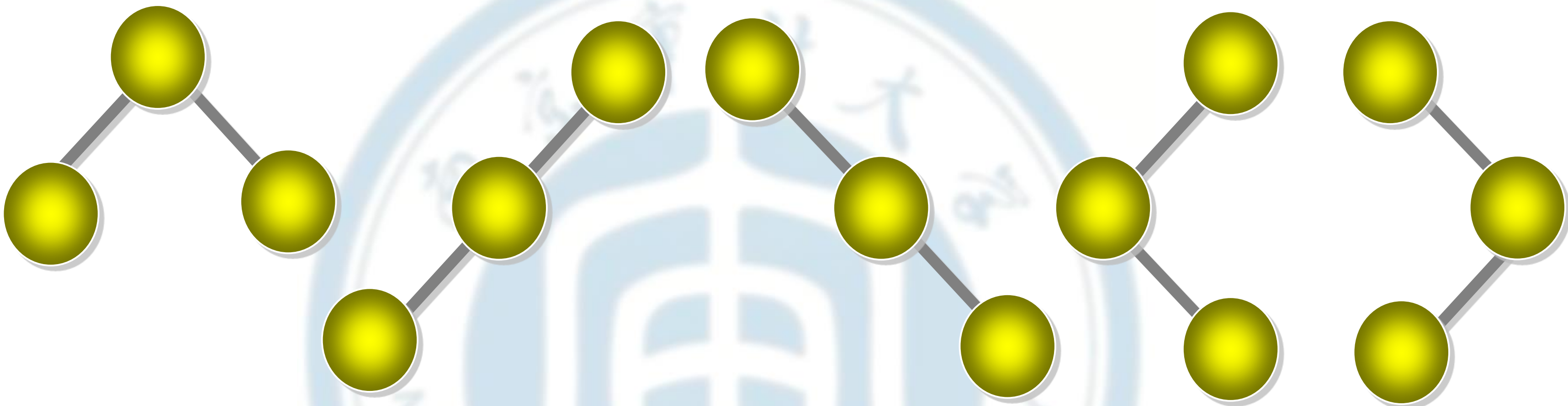
所以，二叉树不是前面定义的树的特殊形式，而是另外一种数据结构。

二叉树的子树有左，右之分，且其次序不能任意颠倒，即使是一个孩子也有左右之分。在有序树中，虽然一个结点的孩子之间是有左右次序的，但是若该结点只有一个子树就无需区分其左右子树

- 二叉树不是树的特例
  - 二叉树的子树有左、右之分，且其次序不能任意颠倒，即使是一个孩子也有左右之分
  - 在有序树中，虽然一个结点的孩子之间是有左右次序的，但是若该结点只有一个子树就无须区分其左右次序
- 基本形态



练习：具有3个结点的二叉树有多少种？



二叉树的性质

- 性质1：在二叉树的第 i 层至多有 2^{i-1} 个结点（ $i\ge1$）。
- 性质2：深度为 k 的二叉树至多有 2^k-1 个结点。
- 性质3：对于任何一棵二叉树T，若其终端结点(叶子)数为 n_0 ，度为1的结点数为n_1，度为2的结点数n_2，则$n_0= n_2+1$。

性质 1：

在二叉树的第 i 层上至多有 2^{i-1} 个结点。(i≥1)

用归纳法证明：

归纳基：$i = 1$ 层时，只有一个根结点，

$2^{i-1} = 2^0 = 1$;

归纳假设：假设对所有的 j，$1 \le j < i$，命题成立;

归纳证明：二叉树上每个结点至多有两棵子树，

则第 i 层的结点数 $= 2^{i-2} \times 2 = 2^{i-1}$ 。

- **性质 2：**

深度为 ***k*** 的二叉树上至多含 $2^k\text{-}1$ 个结点. (k≥1)

证明：

基于上一条性质，深度为 ***k*** 的二叉树上的结点数至多为

$$2^0+2^1+\cdots\cdots+2^{k-1} = 2^k-1$$

- **性质 3：**

对任何一棵二叉树，若它含有n_0个叶子结点、n_2个度为 2 的结点，则必存在关系式：$n_0 = n_2+1$

证明： 设 二叉树上结点总数 $n = n_0 + n_1 + n_2$

又 二叉树上分支总数 $b = n_1 + 2n_2$

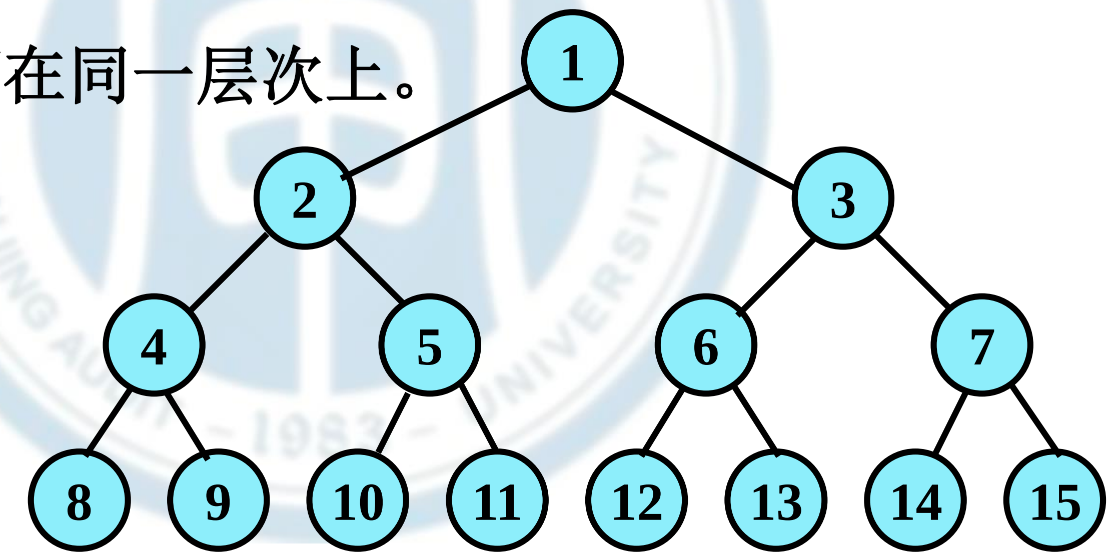
而 $b = n - 1$

由此， $n_1 + 2n_2 = n_0 + n_1 + n_2 - 1$

即 $n_0 = n_2 + 1$

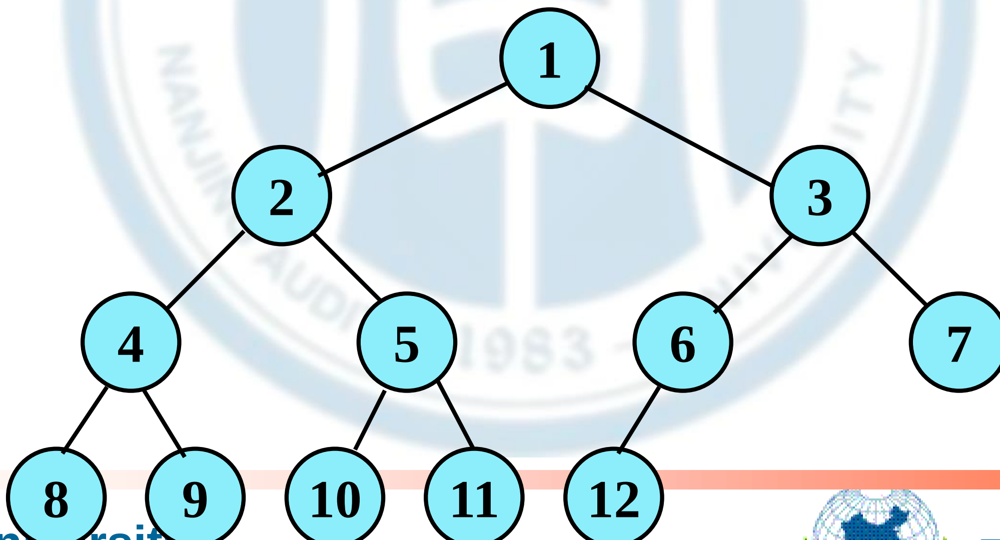
特殊二叉树之满二叉树

- 深度为k且有2^k-1个结点的二叉树。
- 特点
  - 每一层上的结点数都是最大结点数；
  - 所有的分支结点的度数都为2；
  - 叶子结点都在同一层次上。

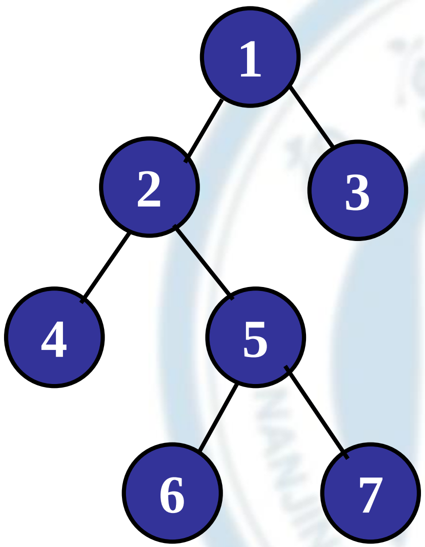


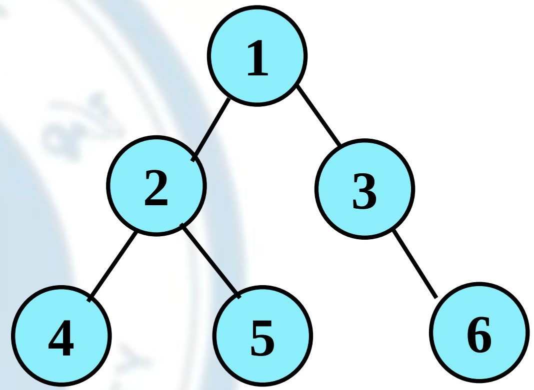
特殊二叉树之完全二叉树

- 若对满二叉树的结点从上到下从左至右进行编号，则深度为k且有n个结点的二叉树称为完全二叉树，当且仅当其每一个结点都与深度为k的满二叉树的编号从1到n一一对应时。
- 特点
  - 叶子结点只可能在层次最大的两层上出现；
  - 前k-1层中的结点都是“满”的，且第 k 层的结点都集中在左边。



判断是否为完全二叉树





- 性质4：具有n个结点的完全二叉树的深度是$\lfloor \log_2 n \rfloor+1$。
- 性质5：如果对一棵有n个结点的完全二叉树的结点按层序编号，则对任一结点i(1≤i≤n)，有：
  - 如果i=1，则结点i是二叉树的根，无双亲；如果i>1，则其双亲是$\lfloor i/2 \rfloor$；
  - 如果2i>n，则结点i无左孩子；如果2i≤n，则其左孩子是2i；
  - 如果2i+1>n，则结点i无右孩子；如果2i+1≤n，则其右孩子是2i+1。

性质 4：具有 *n* 个结点的完全二叉树的深度为 $\lfloor \log_2 n \rfloor + 1$

具有n个结点的完全二叉树的深度为[log2n]+1（向下取整）

证明：

设 完全二叉树的深度为 k

则根据性质2得 $2^{k-1}-1 < n \leq 2^k-1 \Rightarrow 2^{k-1} \leq n < 2^k$

因为 k 只能是整数，因此， $k = \lfloor \log_2 n \rfloor + 1$

- **性质 5：**

若对含 *n* 个结点的完全二叉树从上到下且从左至右进行 *1* 至 *n* 的编号，则对完全二叉树中任意一个编号为 *i* 的结点：

(1) 若 $i=1$，则该结点是二叉树的根，无双亲, 否则，编号为 $\lfloor i/2 \rfloor$ 的结点为其双亲结点;

(2) 若 $2i>n$，则该结点无左孩子，

否则，编号为 $2i$ 的结点为其左孩子结点；

(3) 若 $2i+1>n$，则该结点无右孩子结点，

否则，编号为$2i+1$ 的结点为其右孩子结点。

习题：

设一棵完全二叉树具有1000个结点，则此完全二叉树有___500___个叶子结点，有___499___个度为2的结点，有___1___个结点只有非空左子树，有___0___个结点只有非空右子树。