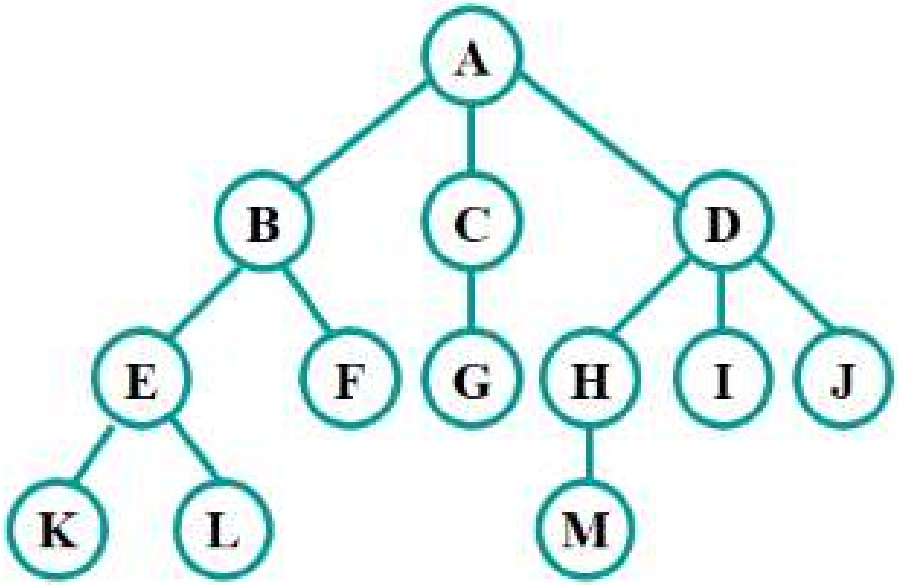
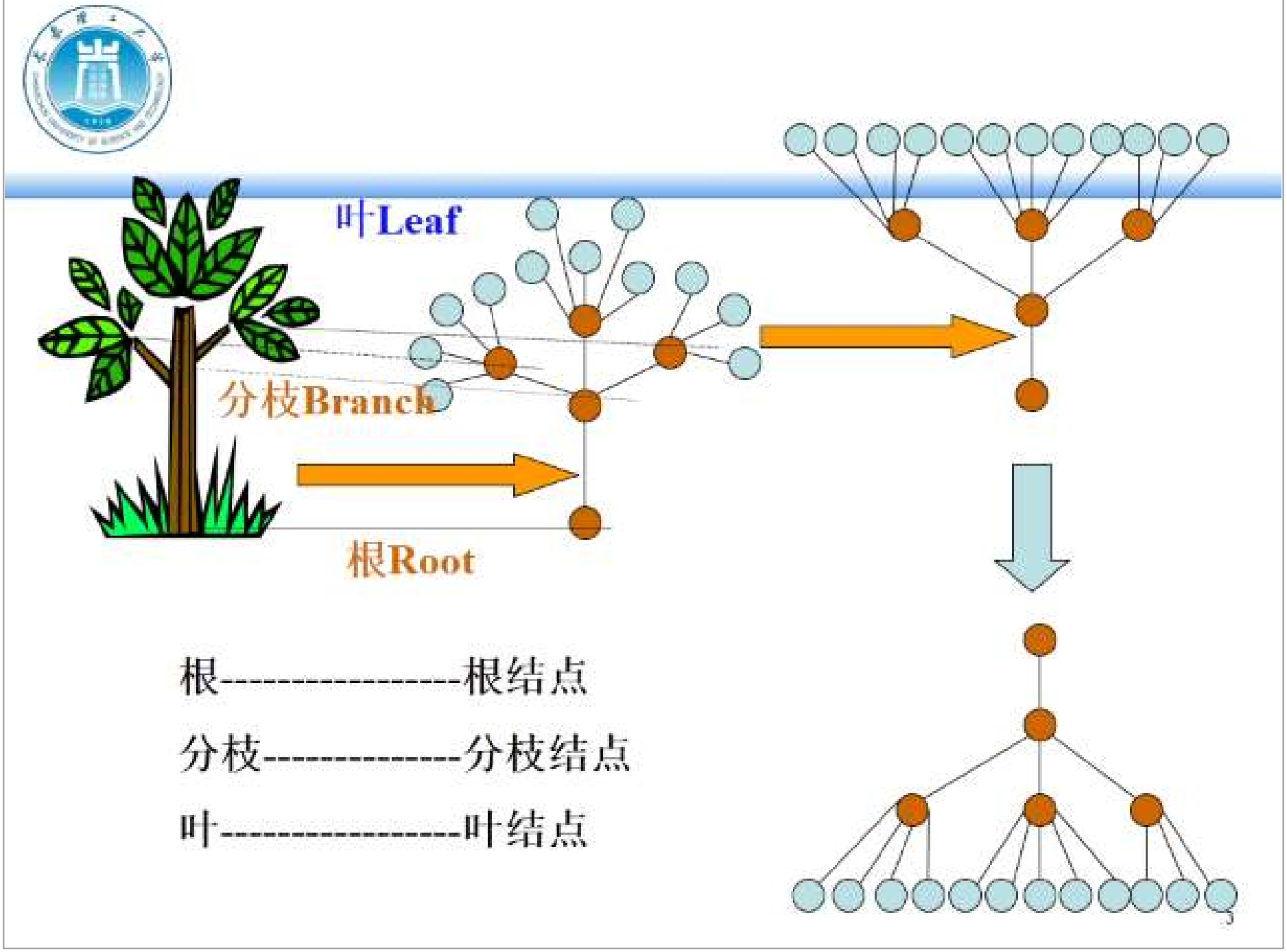
填空题 5分

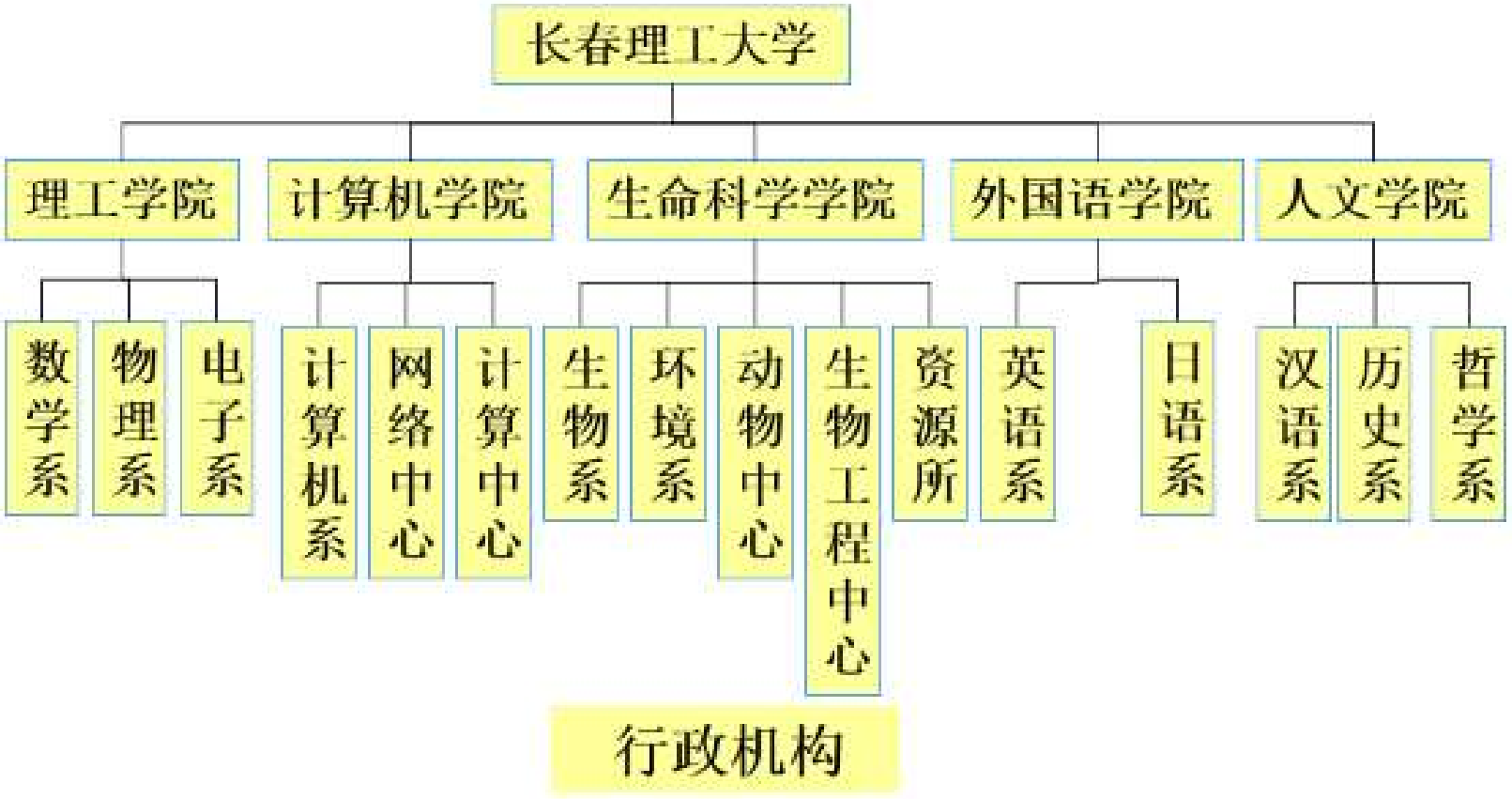
结点B的度 [填空1]、结点I的双亲 [填空2] 、结点B，C，D为 [填空3] 、结点F，G为 [填空4] 、树的深度 [填空5] 。




叶Leaf
分枝Branch
根Root
根----------根结点
分枝----------分枝结点
叶----------叶结点

树形结构是一种非线性结构，应用十分广泛。

如：行政机构、家谱、磁盘目录等



树的基本概念

- 术语主要来源于家谱和树：

双亲、父母（Parent）；子女、孩子（Child）：

若〈a,b〉∈R，则称a是b的双亲，b是a 的子女(孩子).

叶结点（Leaf）：度为0的结点；

分枝结点（Branch node）：度大于0的结点；

结点度（Degree）：该结点所拥有的子女数；

树的度：树内各结点度的最大值；

结点的层次（Level）：设根结点的层次为1，其它任一结点所在的层次是其双亲的层次加1；

树的基本概念

深度（Depth）：树中结点的最大层次；或称为高；

兄弟（Sibling)：具有同一双亲的结点互称兄弟；

堂兄弟（Cousin）：双亲在同一层的结点之间互称堂兄弟；

路径（Path）：如果有结点序列$n_1,n_2,n_3,\ldots,n_k$，并且前1个结点是后1个结点的双亲；它的长度是k-1；

祖先、后代（ancestor）：一个结点是它所有子树中的结点的祖先，这些结点是它的后代(子孙)；

有序树（Ordered tree）：每个结点的子女由左到右是有次序的称有序树；否则是无序树；

树的基本概念

❖树的定义：树是n（n>=0）结点的有限集，

在任意非空树中：

(1) 有且仅有一个特定的结点称为根（Root）的结点.

(2) 当n>1时，其余的结点可分为m个互不相交的子集 $T_1, T_2, \ldots, T_m$，每个子集又都是树，称为根的子树（Sub tree）.

树的定义具有递归性

树的基本概念

❖抽象数据类型树的定义：

ADT Tree{

数据对象**D**：**D**是具有相同特性的数据元素的集合。

数据关系**R**：若**D**为空集，则称为空树；
若**D**仅含一个数据元素，则**R**为空集，否则**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) 若**D-{root}≠** ϕ 则存在**D-{root}**的一个划分**D1**，**D2**，**…**，**Dm**，**(m>0)**，对任意**j≠k(1≤j,k≤m)**有**Dj∩Dk=** ，ϕ 且对任意的**i(1≤i≤m)**，唯一存在数据元素**xi∈Di**，有**<root，xi>∈H**；

树的基本概念

(3) 对应于D-{root}的划分，H-{<root，x_1>，…，<root，x_m>}有唯一的一个划分H_1, H_2，…，H_m(m>0)，对任意j≠k(1≤j，k≤m)有$H_j \cap H_k = \phi$，且对任意i(1≤i≤m)，H_i是D_i上的二元关系，(D_i，{H_i})是一棵符合本定义的树，称为根root的子树。

基本操作：

InitTree(&T)

操作结果：构造空树T。

DestroyTree(&T)

初始条件：树T存在。

操作结果，销毁树T。

树的基本概念

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。

ClearTree(&T)

初始条件：树T存在。

操作结果：将树T清为空树。

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FALSE。

TreeDepth(T)

初始条件：树T存在。

操作结果：返回T的深度。

树的基本概念

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。

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。

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”。

树的基本概念

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”。

树的基本概念

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

初始条件：树T存在,p指向T中某个结点，i指结点的度，
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，i指结点的度，

操作结果：删除T中p所指结点的第i棵子树。

TraverseTree(T);

初始条件：树T存在。

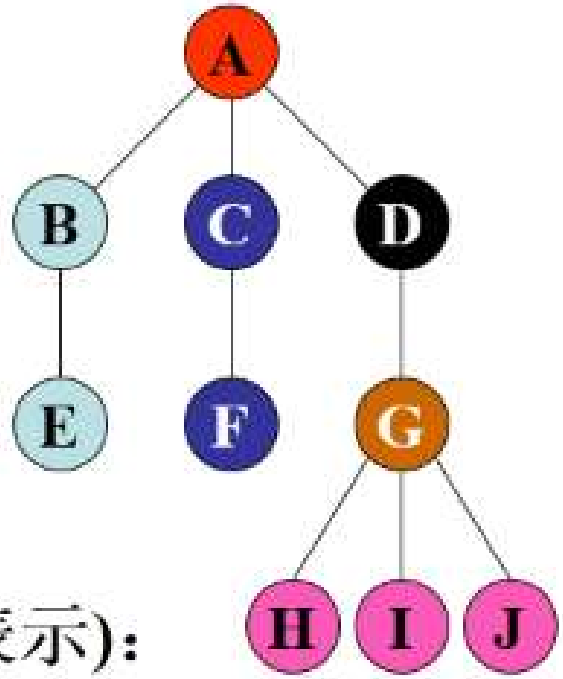
操作结果：按某种次序对T的每个结点进行遍历。

}ADT Tree

6.1 树（Tree）的定义和基本术语

- 树的表示方法：

1. 树形表示：



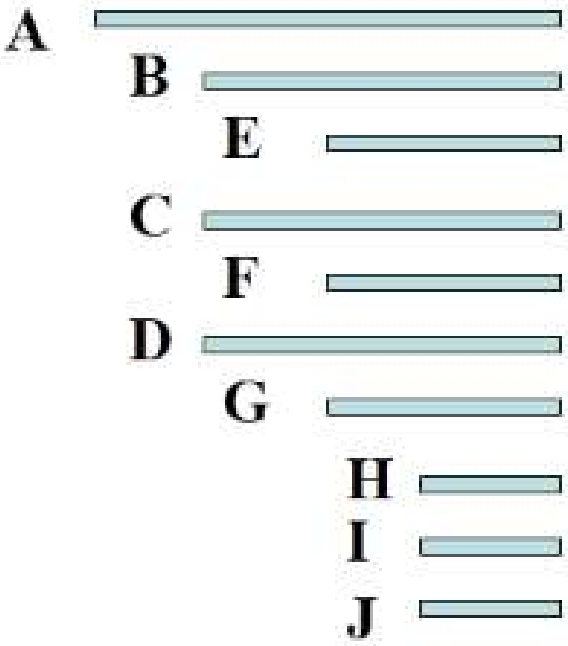
2. 括号表示(广义表表示)：

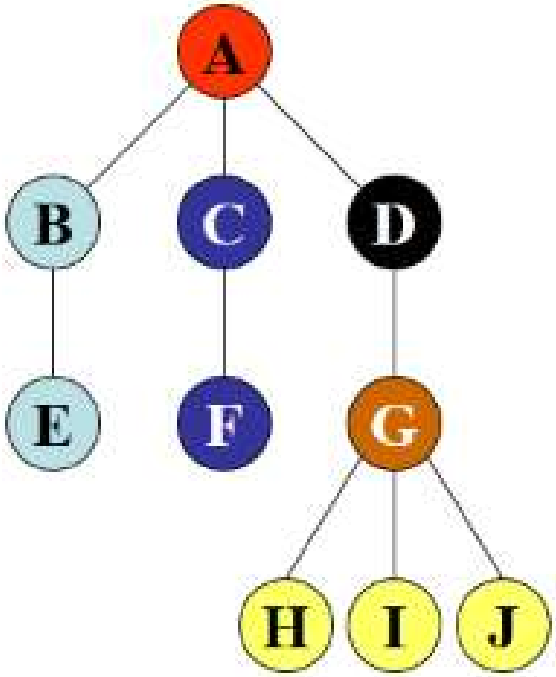
(A(B(E))(C(F))(D(G(H)(I)(J))))

树根 T_1 T_2 T_3

6.1 树（Tree）的定义和基本术语

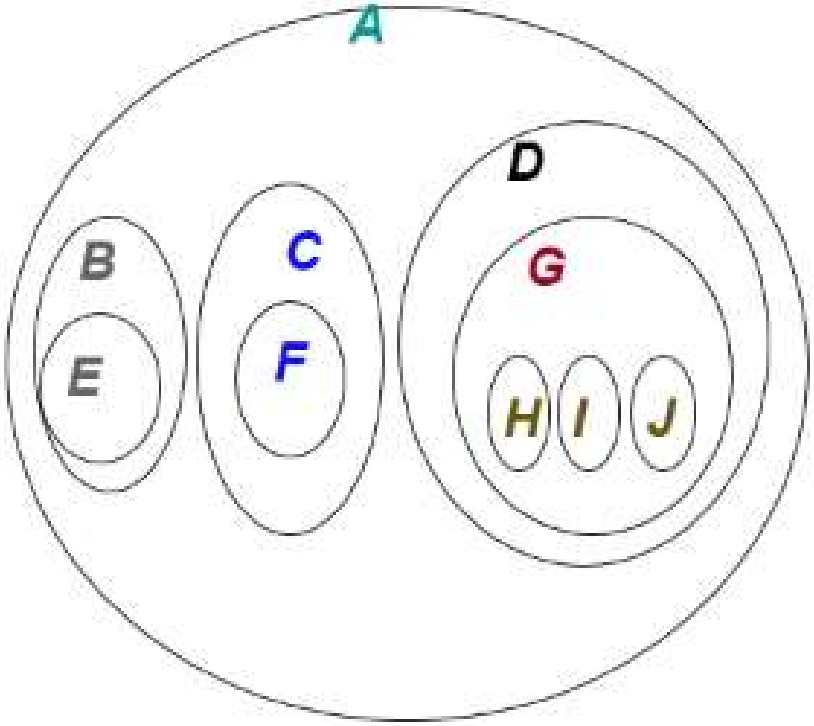
3. 凹入表表示(目录表示法):

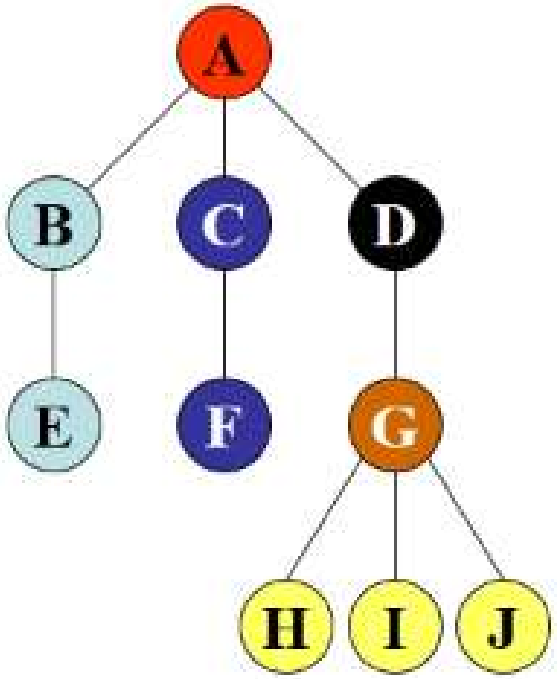




6.1 树（Tree）的定义和基本术语

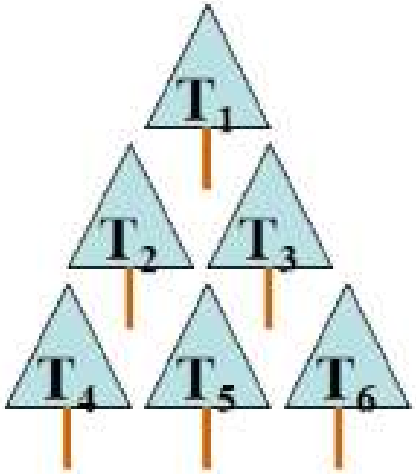
4. 文氏图表示(集合表示):



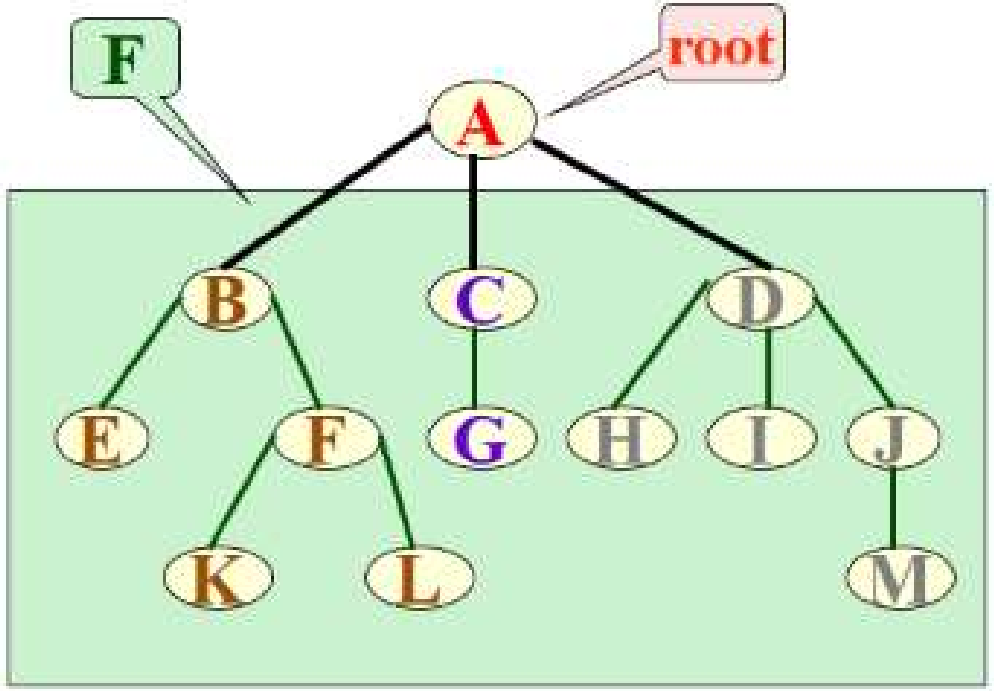


树的基本概念

❖森林（Forest）：是m(m≥0)棵互不相交的树的集合



(例如删除树根后的子树构成一个森林)



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

二叉树的基本概念

6.2.1 二叉树的定义

二叉树：是n（n>=0）结点的有限集，在任意非空树中：

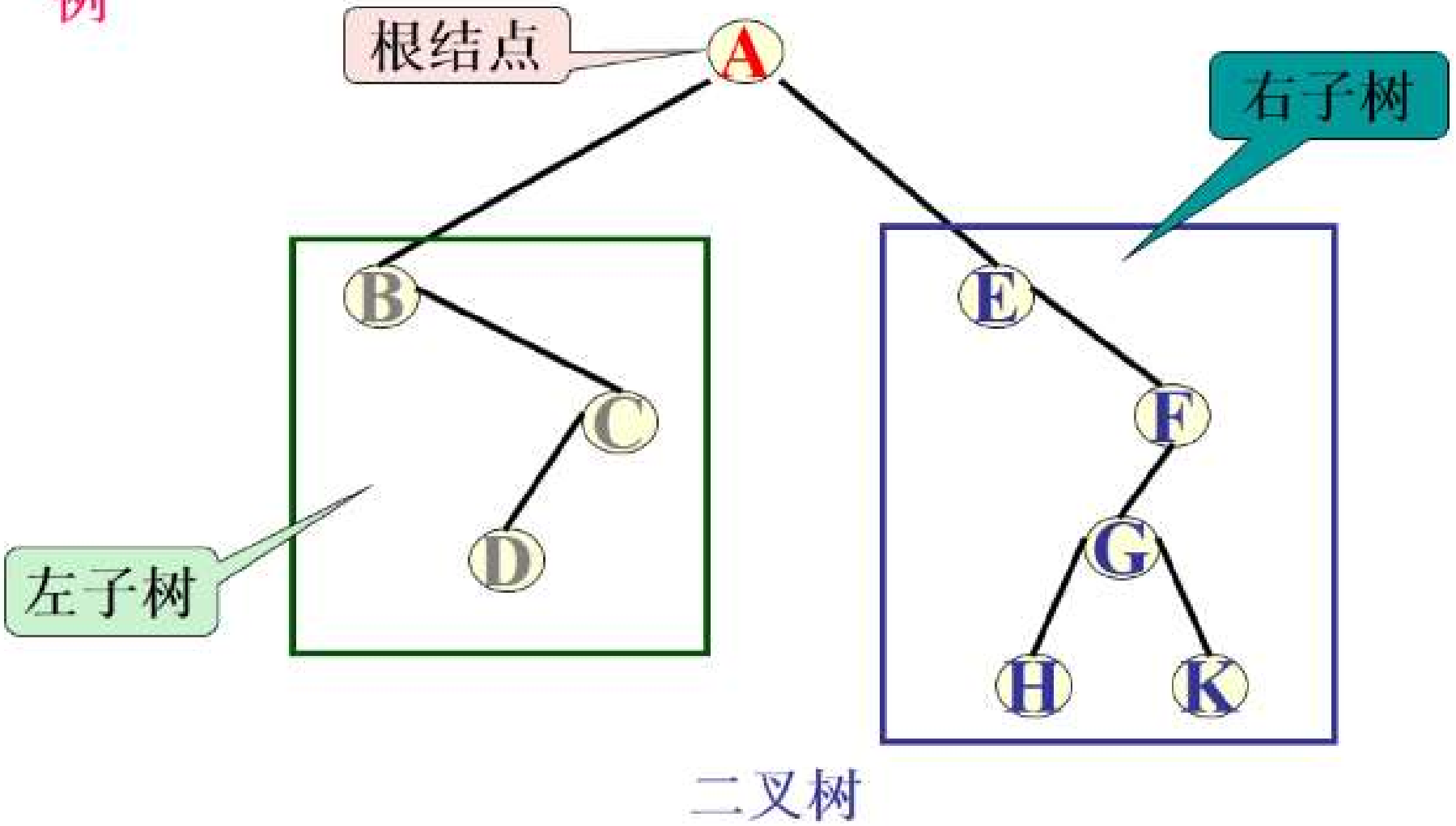
(1) 有且仅有一个特定的称为根（Root）的结点；

(2) 当n>1时，其余的结点最多分为2个互不相交的子集 T_1，T_2，每个又都是二叉树，分别称为根的左子树和右子树.

二叉树是另一种树形结构。

二叉树的基本概念

例



投票 最多可选1项

具有3个结点的二叉树可能有几种不同形态？

A 1

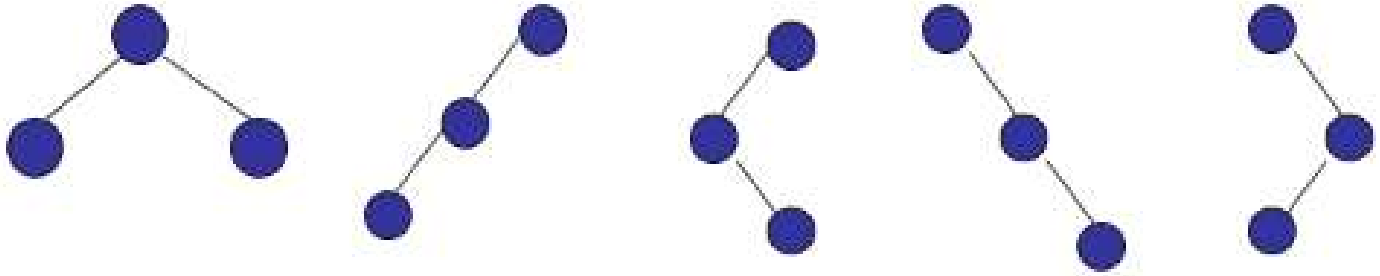
B 2

C 3

D 5

二叉树的基本概念

具有3个结点的二叉树可能有几种不同形态？



二叉树的性质

性质1: 在二叉树的第i层最多有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：　i=k时命题成立；

归纳证明：　i=k+1时,二叉树上每个结点至多有两棵子树，

则第 i 层的结点数

$= 2^{k-1} \times 2 = 2^{(k+1)-1} = 2^{i-1}$ 。

二叉树的性质

性质2：深度为k的二叉树最多有2^k-1个结点（k>=1).

证明：

基于性质1，深度为 k 的二叉树上的结点数最多为

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

二叉树的性质

性质3: 任一二叉树，若叶结点数是n_0，度为2的结点数是 n_2 ,则 $n_0 = n_2 + 1$

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 $b = n-1 = n_0 + n_1 + n_2 - 1$

由此， $n_0 = n_2 + 1$