第6章　树与二叉树

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
6.5　哈夫曼树及其应用

6.1 树的定义和基本术语

定义:

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 。
- 否则:

  (1) 在D中存在唯一的称为根的数据元素root;

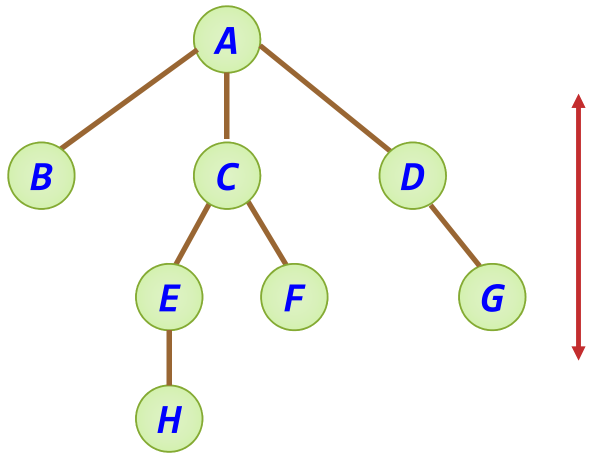
  (2) 当n>1时，其余结点可分为m (m>0)个互不相交的有限集$T_1, T_2, \ldots, T_m$，其中每一个子集本身又是一棵符合本定义的树，称为根root的子树。

从数据结构角度看，树包含n（$n \geq 0$）个结点，当$n=0$时，称为空树；非空树的定义如下：

$$T=(D, R)$$

其中，D为树中结点的有限集合，关系R满足以下条件：

- 有且仅有一个结点$d_0 \in D$，它对于关系R来说没有前驱结点，结点d_0称作树的根结点。
- 除根结点d_0外，D中的每个结点有且仅有一个前驱结点，但可以有多个后继结点。
- D中可以有多个终端结点。



【示例-1】 有一棵树T=(*D*，*R*)，其中

D={*A*，*B*，*C*，*D*，*E*，*F*，*G*，*H*}，

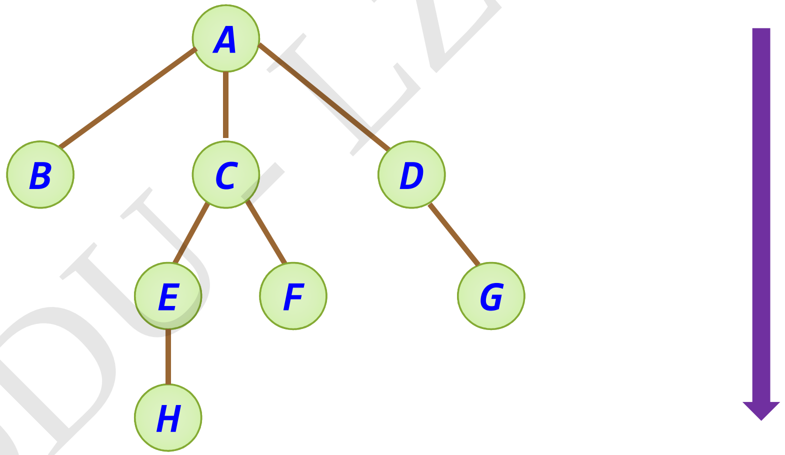
R={r}

r={<*A*,*B*>，<*A*,*C*>，<*A*,*D*>，<*C*,*E*>，<*C*,*F*>，<*D*,*G*>，<*E*,*H*>}，画出其逻辑结构图。

解：A是根结点，其余结点分成三个互不相交的子集：

T_1={B}，T_2={C，E，F，H}，T_3={D，G}。

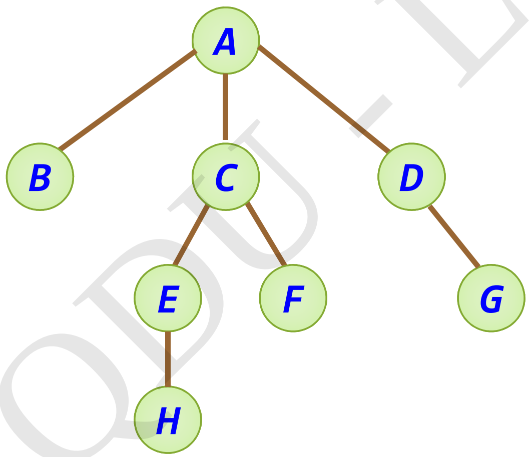
T_1、T_2、T_3都是根结点A的子树，且各自本身也是一棵树。



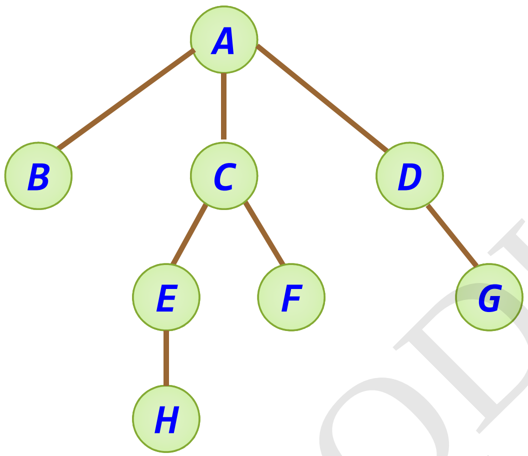
- 说明：树中结点之间的关系应为有向关系，在上图中，结点之间的连线即分支线都是有向的，默认箭头向下的。

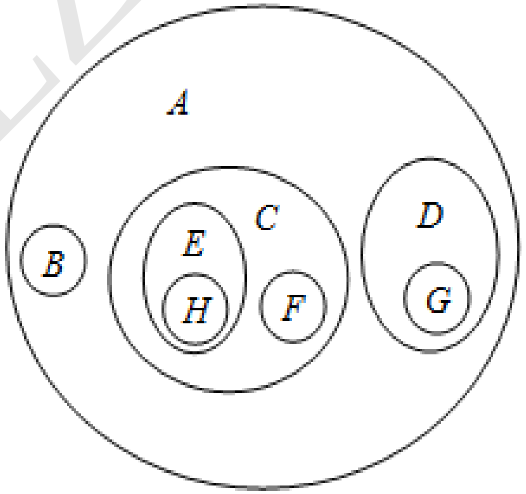
树的表示法

- **树形表示法**。这是树的最基本的表示，使用一棵倒置的树表示树结构，非常直观和形象。

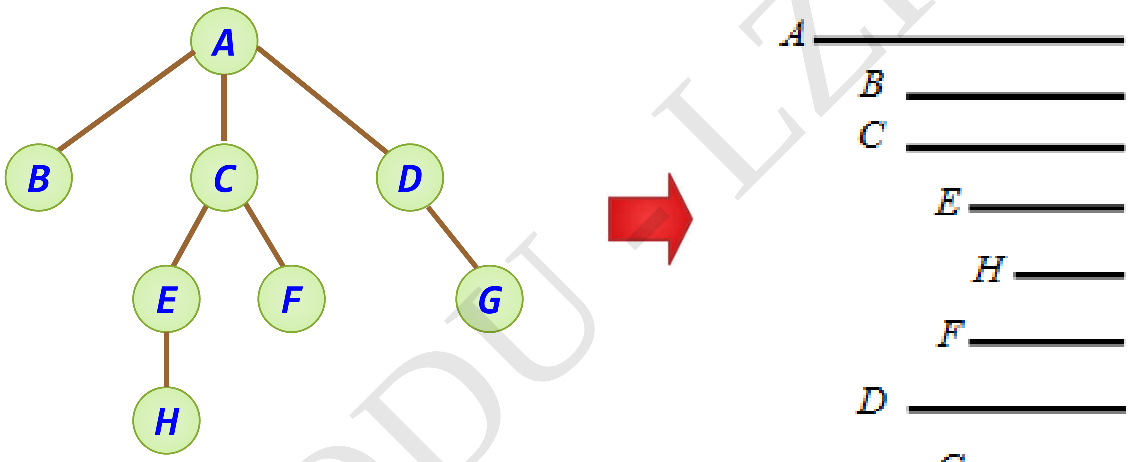


- **文氏图表示法。**使用集合以及集合的包含关系描述树结构。

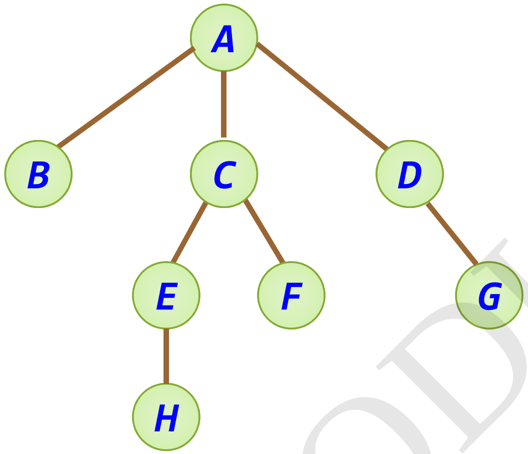




□ **凹入表示法。** 使用线段的伸缩关系描述树结构。



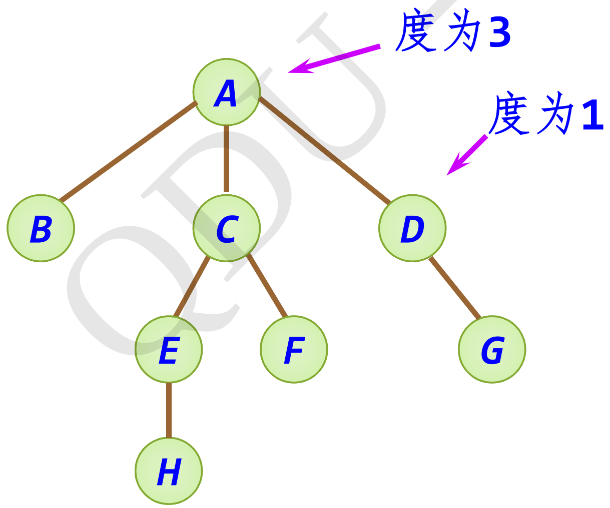
- **广义表表示法。**将树的根结点写在括号的左边，除根结点之外的其余结点写在括号中并用逗号分隔。



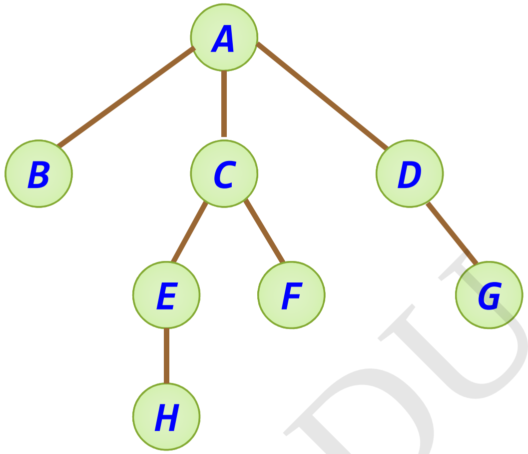
A(B,C(E(H),F),D(G))

树的基本术语

- **树的结点。**包含一个数据元素及若干指向其子树的分支。
- **结点的度。**树中每个结点具有的子树数或者后继结点数称为该结点的度。

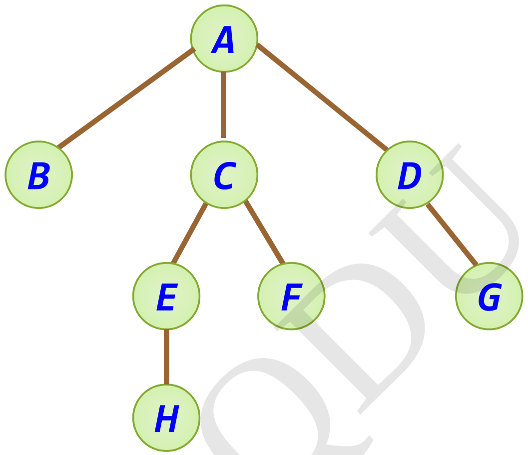


- **树的度**。树中所有结点的度的最大值称之为树的度。



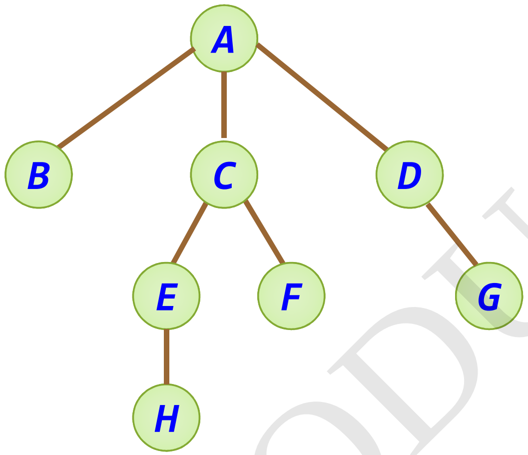
树的度为3

- **分支结点**。度大于0的结点称为分支结点或非终端结点。度为1的结点称为单分支结点，度为2的结点称为双分支结点，依次类推。



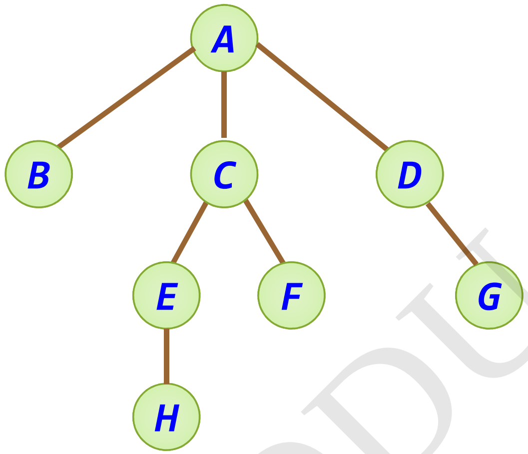
A、C、D、E为分支结点

➢ **叶子结点（或叶结点）**。度为零的结点称为叶子结点或终端结点。



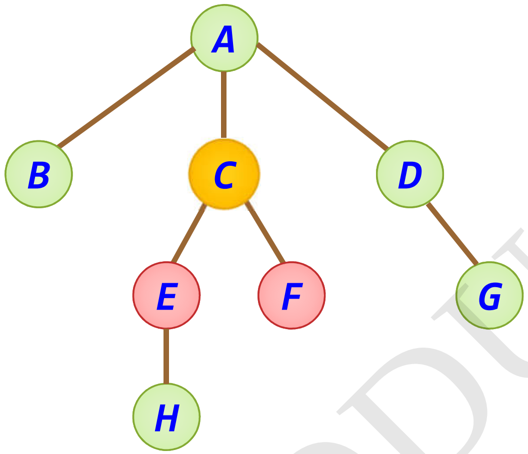
***B、H、F、G*为叶子结点**

- **孩子结点**。一个结点的后继称之为该结点的孩子结点。



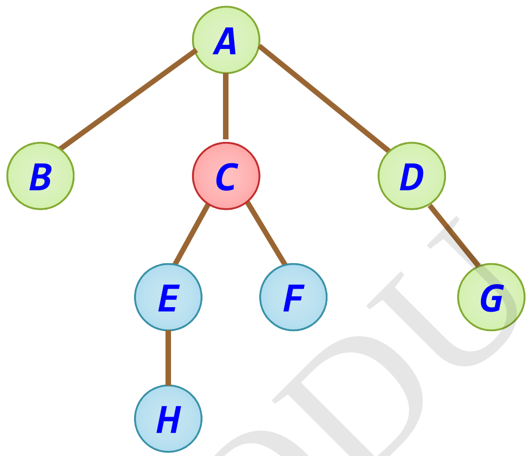
结点*A*的孩子结点为*B*、*C*和*D*

- **双亲结点（或父亲结点）**。一个结点称为其后继结点的双亲结点。



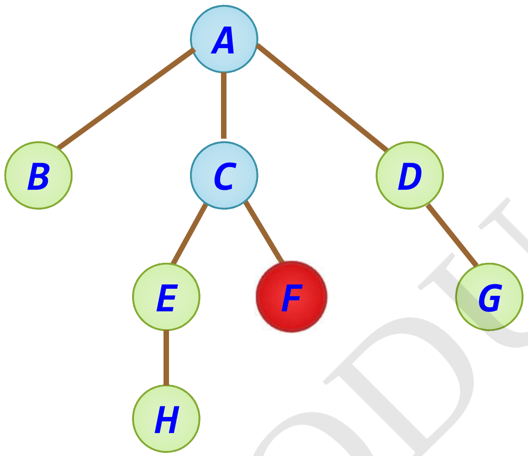
结点*E*和*F*的双亲结点均为*C*

- **子孙结点**。一个结点的子树中除该结点外的所有结点称之为该结点的子孙结点。



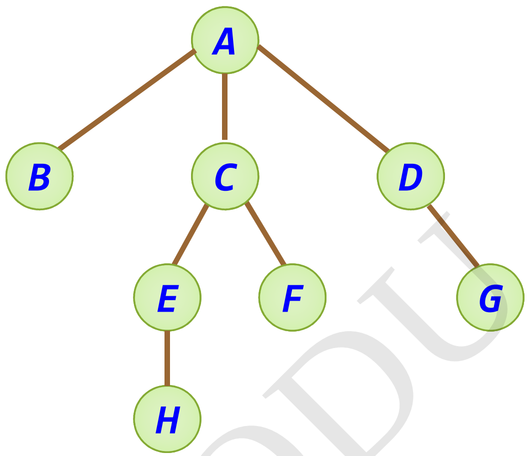
结点*C*结点的子孙结点为*E*、*F*和*H*

- **祖先结点**。从树根结点到达某个结点的路径上通过的所有结点称为该结点的祖先结点（不含该结点自身）。



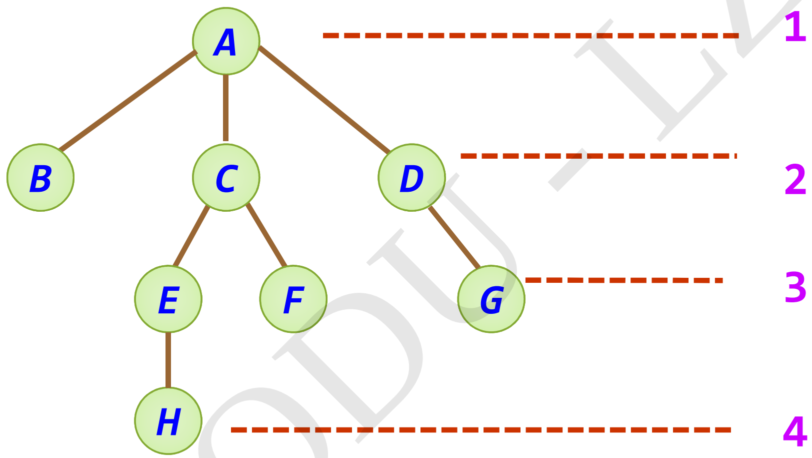
结点*F*的祖先结点为*A*、*C*

➢ **兄弟结点**。具有同一双亲的结点互相称之为兄弟结点。

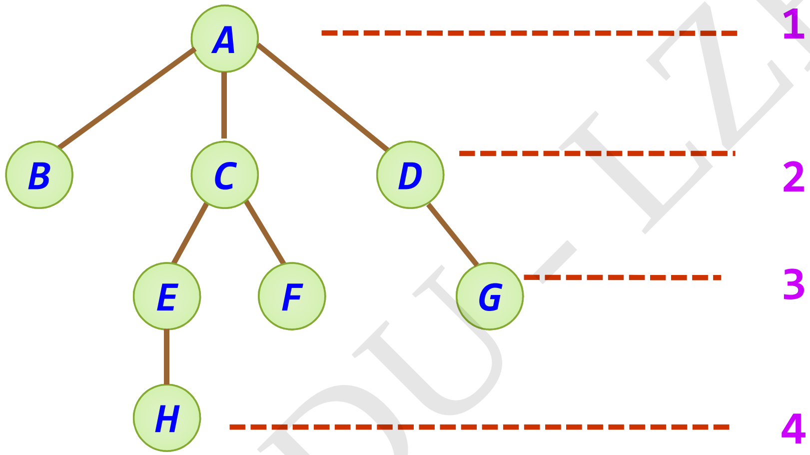


结点*E*和*F*是兄弟结点

- **结点层次**。树具有一种层次结构，根结点为第一层，其孩子结点为第二层，如此类推得到每个结点的层次。

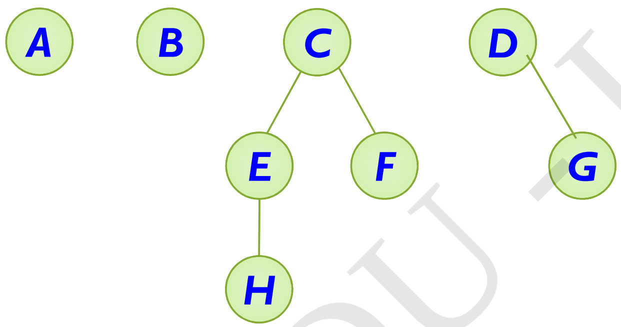


➢ **树的高度**。树中结点的最大层次称为树的高度或深度。



高度是4

➢ **森林**。零棵或多棵互不相交的树的集合称为森林。



4棵树构成的森林

- **有序树**。子树之间存在确定的次序关系。
- **无序树**。子树之间不存在确定的次序关系。

END