根深叶茂德馨，人生巍峨树立

第12讲 树与森林

信息学院（智能应用研究院）
欧 新 宇

建议课时：2

第5章 树和二叉树

考核要点

- **考核大纲**

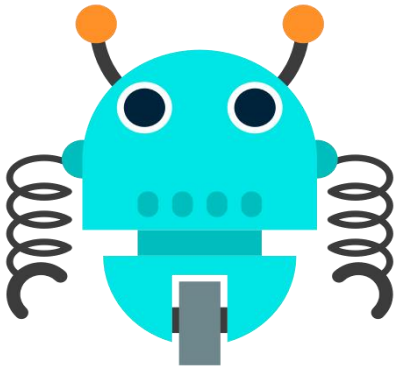
树的基本概念及存储结构；二叉树的定义、主要特征及其存储结构，二叉树的遍历，线索二叉树的基本概念和构造；森林与二叉树的的转换；树和森林的遍历；哈夫曼树和哈夫曼编码。

- **复习要点**
  - 掌握树和二叉树的性质、遍历操作、存储结构和操作特性
  - 了解满二叉树、完全二叉树、线索二叉树、哈夫曼树的定义、性质和思想
  - 熟练掌握二叉树的前、中、后序遍历方法及算法
  - 熟练掌握哈夫曼树的实现方法及构造哈夫曼编码的方法
  - 掌握森林和二叉树的转换方法

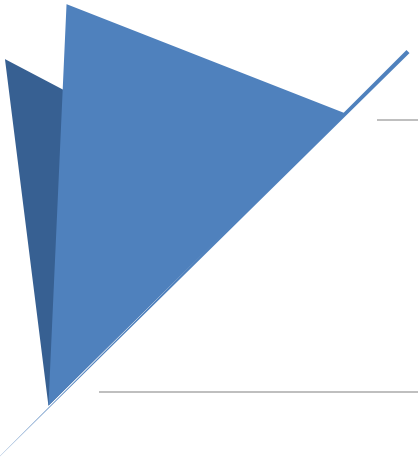
第12讲 树与森林

本讲作业

- **必做题**
  - ✓ 【课堂互动12.1】森林与二叉树的转换
  - ✓ 【课堂互动12.2】树和森林的遍历
  - ✓ 【课后作业12】树与森林
  - ✓ 【课后作业12】[项目005] 树的的综合应用
- **选做题**
  - ✓ 【课前自测12】树与森林
  - ✓ 【扩展练习12】树与森林



- 树的存储结构
- 森林与二叉树的转换
- 树和森林的遍历



树的存储结构

/ 树的抽象数据类型

/ 树的链式存储（多重链表、孩子兄弟、三重链表）

/ 树的顺序存储（双亲表示、带右链的先序表示）

树的存储结构

树的抽象数据类型定义

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R:

1. 若D=∅，则R=∅，则称为空树。
2. 若D≠∅，则R={H}，H具有如下二元关系。
   1). 在D中存在唯一的称为根的数据元素root，它在关系H中无前驱；　　// 关于根的说明
   2). 若D-{root}=∅，则存在D-{root}={D_1, D_r}，且$D_1 \cap D_r = \emptyset$；　　// 关于树不相交的说明
   3). ...　　// 关于子树的说明

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R:

基本操作P:

InitTree(&T)：构造空树T

CreateBiTree(&T, definition)：按definition构造树

TreeEmpty(T)：判断树T是否为空

Assign(T, e, value)：将树T的结点e赋值为value

LeftChild(T, e)：返回树T中结点cur_e的左孩子结点

RightSibling(T, e)：返回树T中结点cur_e的右兄弟结点

...

InsertChild(&T, p, i, c)：插入c为T中p指结点的第i棵树

TraverseTree(T)：按某种次序对T的每个结点访问一次

} **ADT Tree**

树的存储结构

树的两种存储表示方式

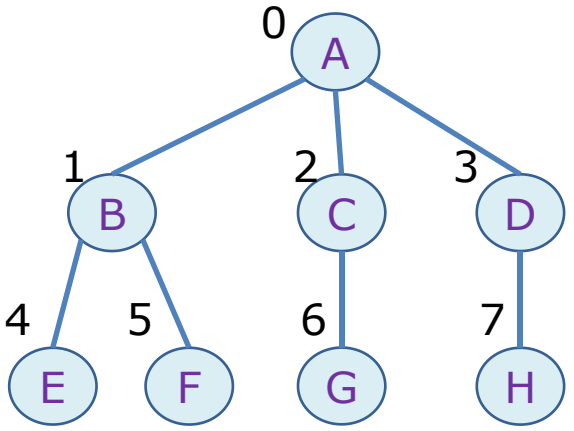
- **基于链接结构的存储表示方法**
  - 多重链表表示法（孩子表示法）
  - 孩子兄弟表示法
  - 三重链表表示法
- **基于顺序结构的存储表示方法**
  - 双亲表示法
  - 带右链的先序表示法

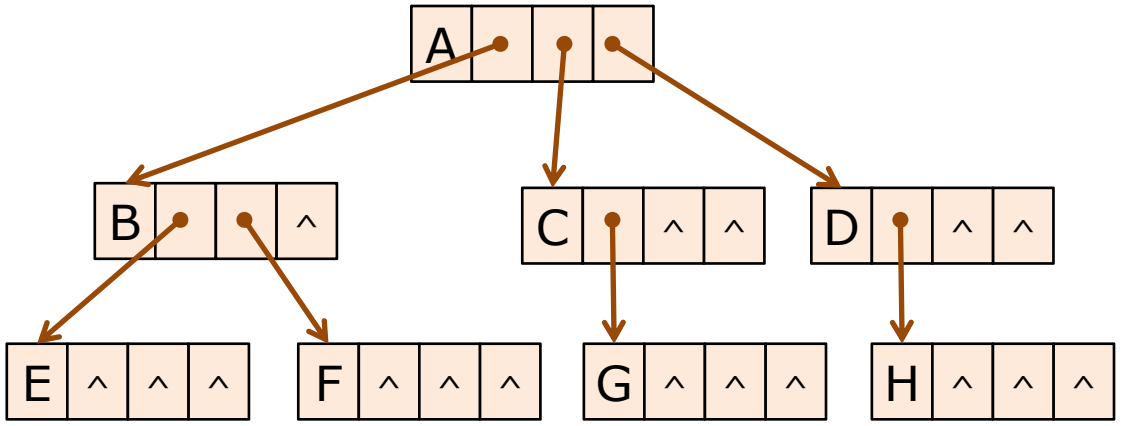
基于链接结构的存储表示方法

多重链表表示法（孩子表示法）

- **存储规则**：设树的度为m，树的每个结点中包含一个数据域和m个指针域
- **结点结构**：

element	$child_1$	$child_2$	...	$child_4$





O(1)+O(d)

- **优点**：结构简单，处理简单
- **缺点**：空指针域多，存在空间浪费

上例中，指针域=结点数*指针域数=8*3=24；
空指针域=17，空间浪费率=17/24=71%。

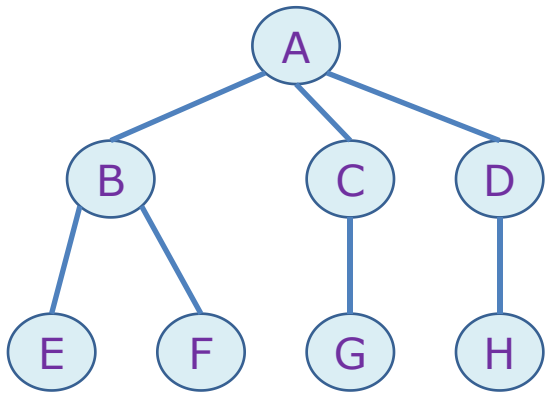
基于链接结构的存储表示方法

孩子兄弟表示法

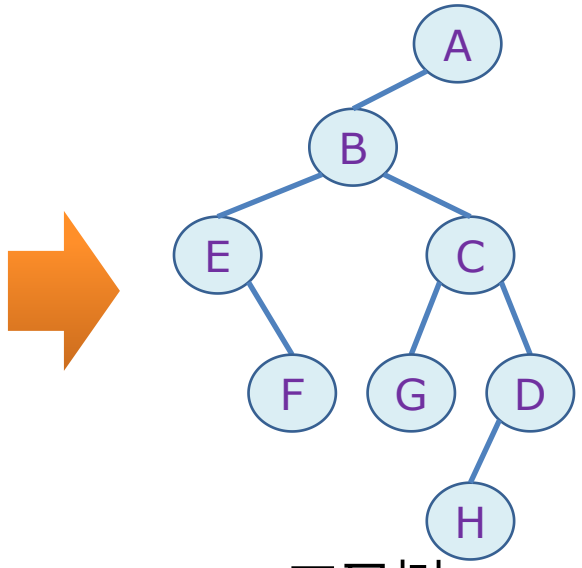
- **存储规则：**树的**孩子兄弟表示法**即森林所对应的二叉树的二叉链表表示法。
- **结点结构：**

LeftChild	element	RightSibling

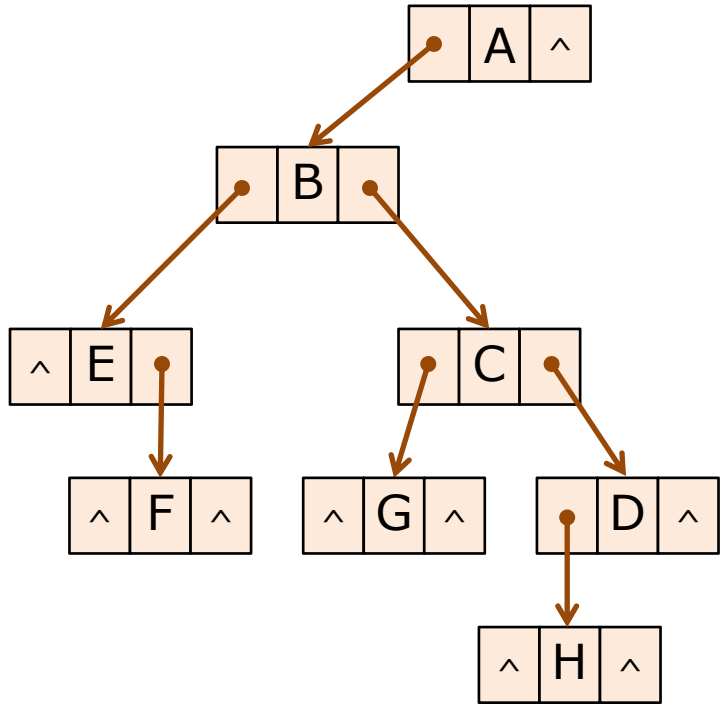
  - ✓ LeftChild：指向当前结点的第一个孩子结点
  - ✓ RightSibling：指向当前结点的第一个兄弟结点



树



二叉树

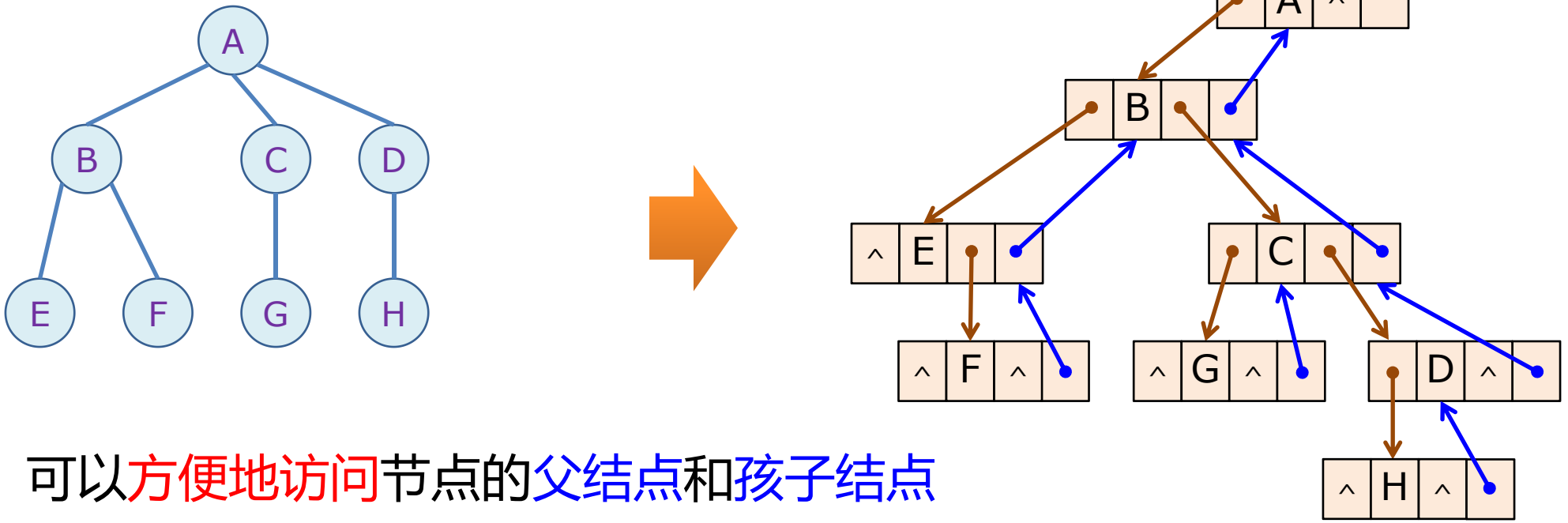


基于链接结构的存储表示方法

三重链表表示法

- **存储规则：**在**孩子兄弟表示法**的基础上，每个结点都增加一个parent指针域，用于指向结点的父结点
- **结点结构：**

LeftChild	element	RightSibling	Parent



- **优点：**可以方便地访问节点的父结点和孩子结点

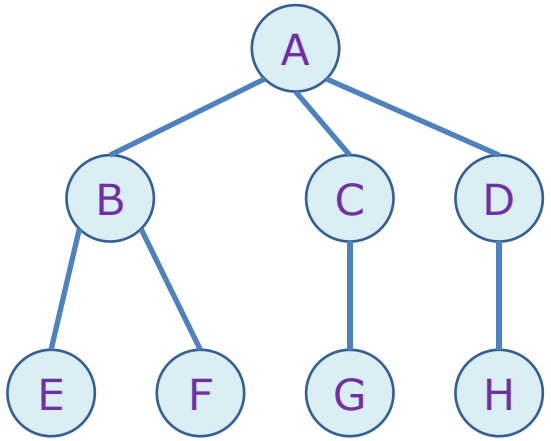
基于顺序结构的存储表示方法

双亲表示法

- **存储规则：**每个结点包含两个域：Element和Parent，其中Parent存储当前结点的父结点的位置下标。顺序表按树中结点层次遍历的次序依次顺序存储。
- **结点结构：**

Element	Parent

所有结点最多只有一个父结点



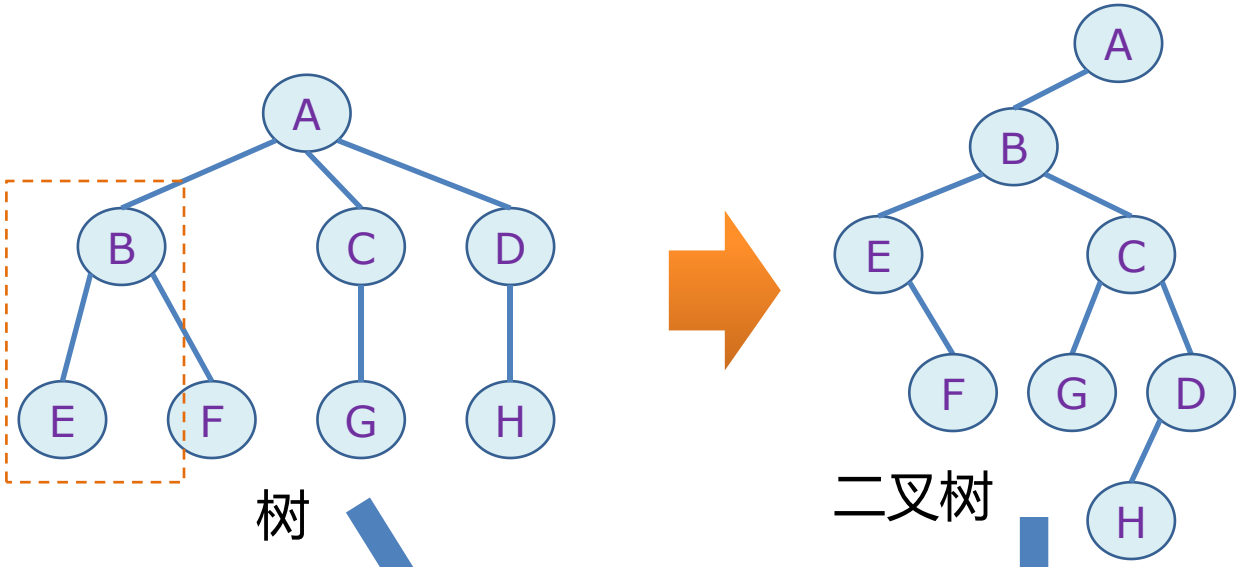
	Element	Parent
0	A	-1
1	B	0
2	C	0
3	D	0
4	E	1
5	F	1
6	G	2
7	H	3

- **优点：**便于查找父结点
- **缺点：**查找孩子结点代价相对较高

基于顺序结构的存储表示方法

带右链的先序表示

- **存储规则：**结点存储顺序按照对应的二叉树的先序遍历的顺序存储



- **结点结构：**

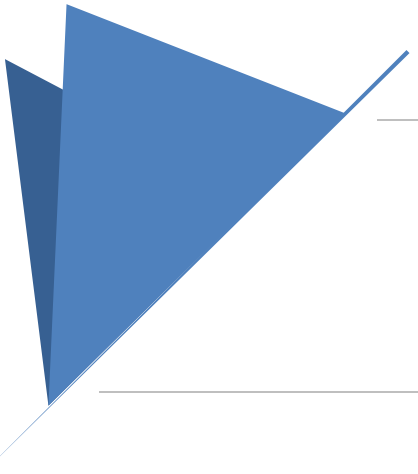
Sibling
Element
iTag

→ 存储兄弟结点的位置下标

→ 存储数据元素

→ 表示当前结点的子结点状态。
0：后继结点为当前结点的子结点；
1：当前结点不存在子结点

	0	1	2	3	4	5	6	7
Sibling	-1	4	3	-1	6	-1	-1	-1
Element	A	B	E	F	C	G	D	H
iTag	0	0	1	1	0	1	0	1



森林与二叉树的转换

/ 树转换为二叉树

/ 森林转换为二叉树

/ 二叉树转换为树

/ 二叉树转换为森林

森林与二叉树的转换

为什么要转换？

不难证明，所有树和森林都能转换成唯一对应的二叉树，不失一般性。

森林与二叉树的转换

森林与二叉树的转换方法

- **森林转换为二叉树**

设 F = (T_1,T_2,...,T_n) 是森林，若F对应的二叉树记为B，则：

1. 若F为空，则B为空二叉树；

2. 若F非空，则B的根是F中第一棵子树T_1的根R_1，B的左子树是R_1的子树森林(T_{11},T_{12},...,T_{1m})所对应的二叉树，B的右子树是森林(T_2,...,T_n)所对应的二叉树。

- **二叉树转换为森林**

设 B = (R,LB,RB)是二叉树，其中R为根，LB为左子树，RB为右子树，森林记为F，设F=(T_1,T_2,...,T_n)，则：

1. 若B为空，则F为空森林；

2. 若B非空，则F的第一棵树T_1的根是二叉树的根R，T_1的根的子树森林是B的左子树LB所对应的森林，F中的其余树(T_2,...,T_n) 是B的右子树RB所对应的森林。

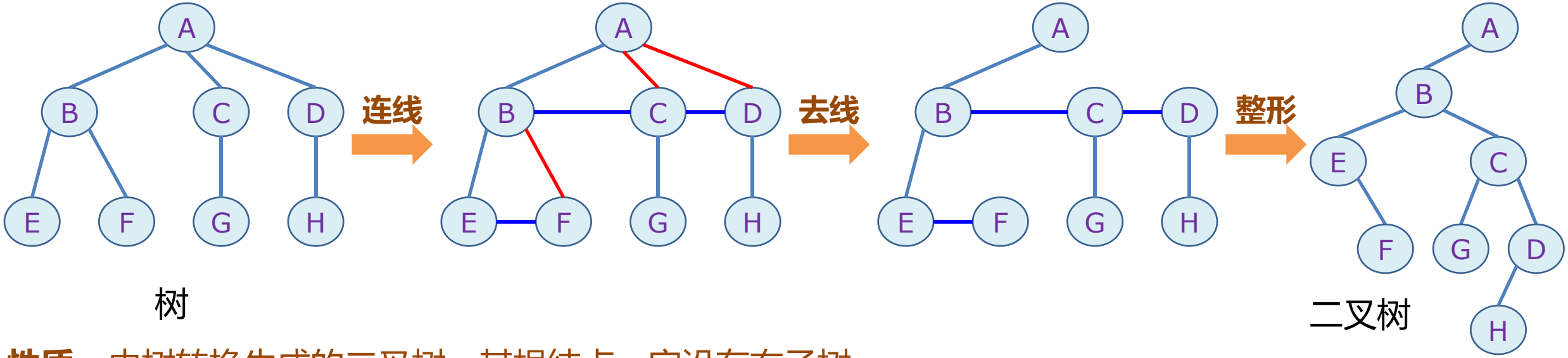
转换方法偏理论化
实际操作存在一定困难

森林与二叉树的转换

树转换为二叉树（长子长兄）

转换步骤：

1. **连线：**将树中具有兄弟关系的结点从左到右依次连线；
2. **去线：**对于每个结点，仅保留双亲与其第一个孩子的连线，并删除其他孩子与双亲之间的连线；
3. **整形：**将兄弟间的连线调整为标准形状的二叉树（根结点+左子树+右子树）。



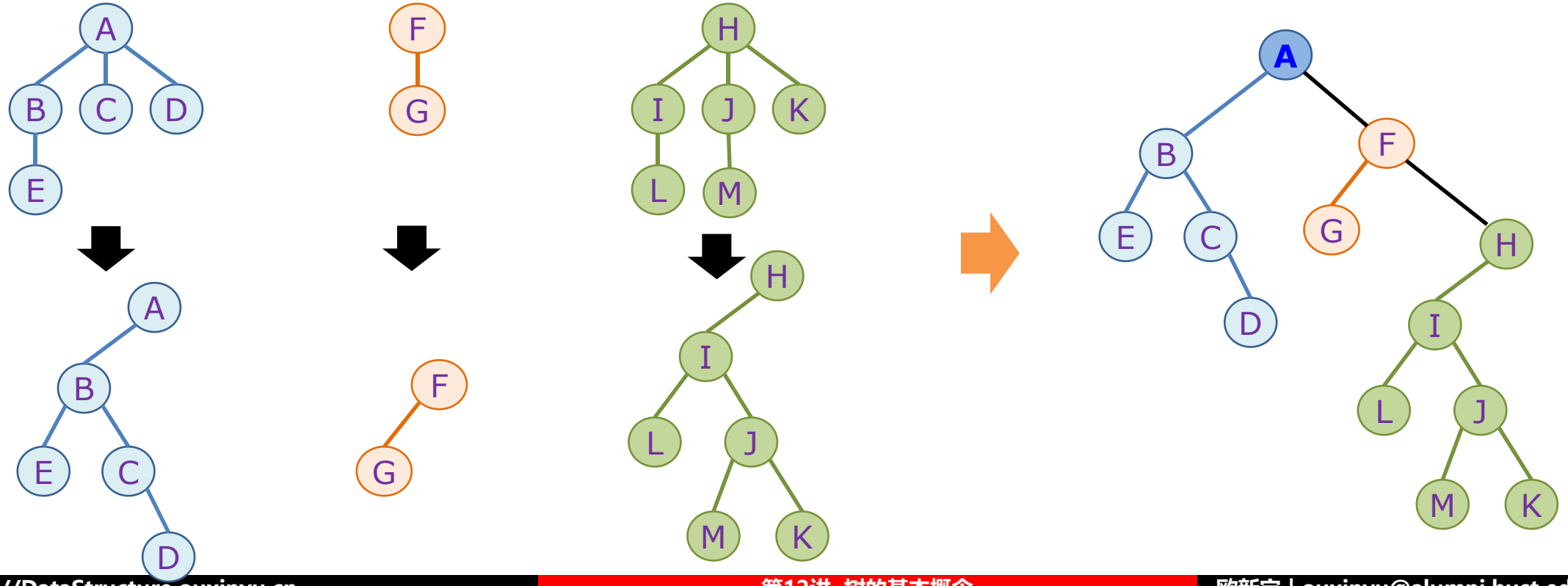
性质：由树转换生成的二叉树，其根结点一定没有右子树。

森林与二叉树的转换

森林转换为二叉树

转换步骤：

1. **转换：**将森林中的每一棵树都转换为对应的二叉树(B_1, B_2,..., B_n)；
2. **组合：**第一棵树不动，从第二棵树开始，将后一棵树的根结点做为前一棵树的右子树并入**总二叉树。**

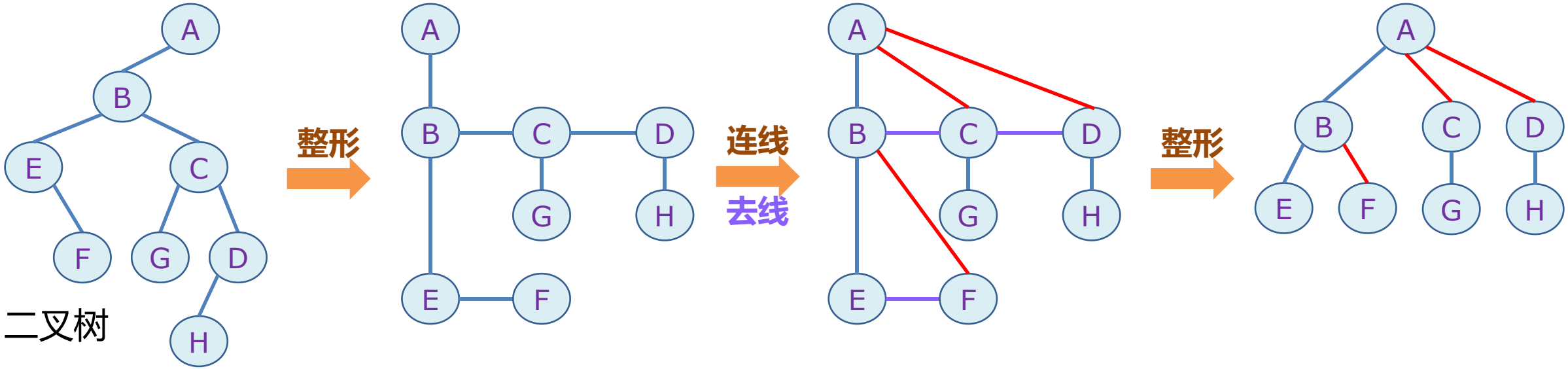


森林与二叉树的转换

二叉树转换为树（右孩子变兄弟）

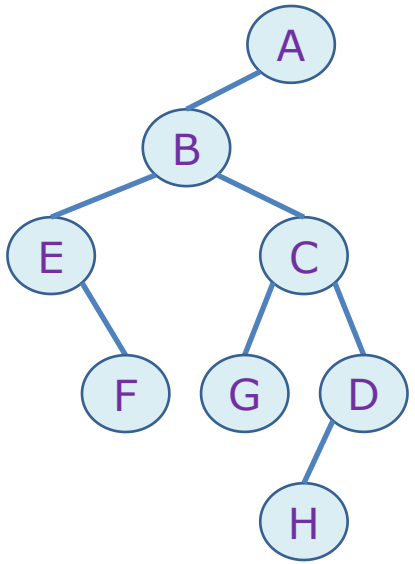
转换步骤：

1. **整形：**将二叉树中所有结点的左孩子连线调整为垂直形态，右孩子调整为水平形态；
2. **连线：**对于任意水平线上的结点，将第一个结点的父节点作为该水平线上所有其他结点的父节点；
3. **去线：**删除所有水平连线，即删除原二叉树中各结点与其右孩子的连线；
4. **整形：**调整步骤3结束之后的树的形状，使其符合树的一般特征要求（父子、兄弟关系清晰）。



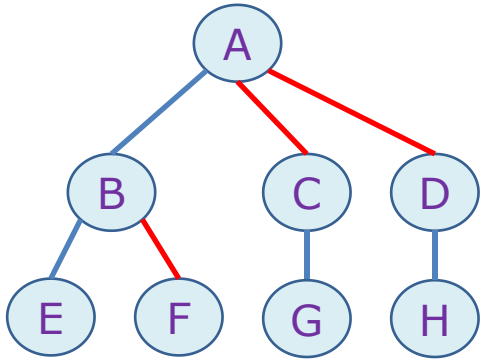
森林与二叉树的转换

二叉树转换为树（右孩子变兄弟）



二叉树

二叉树转换为树



树

若根结点存在右子树，将转换成什么？

性质：

- ✓ 当二叉树的根结点不存在右子树时，该二叉树可以转换成唯一的一棵树；
- ✓ 二叉树的左孩子，在转换之后的树中，依然保持与原父节点之间的父子关系；
- ✓ 二叉树的右孩子，在转换之后的树中，将与原父结点构成兄弟关系。

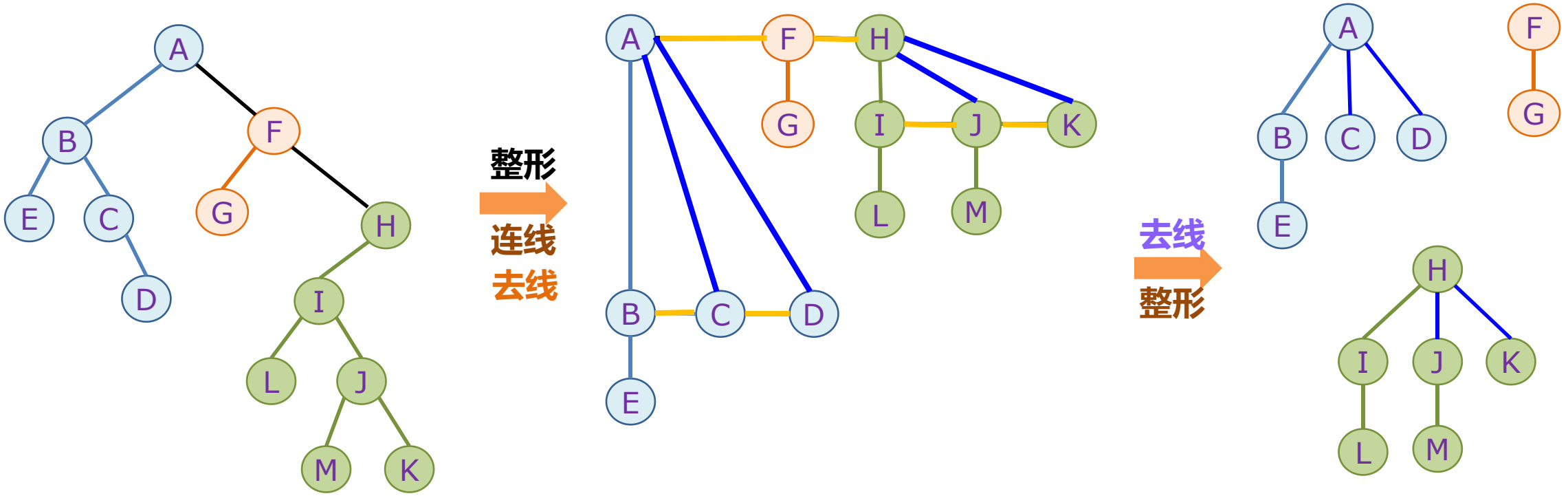
森林与二叉树的转换

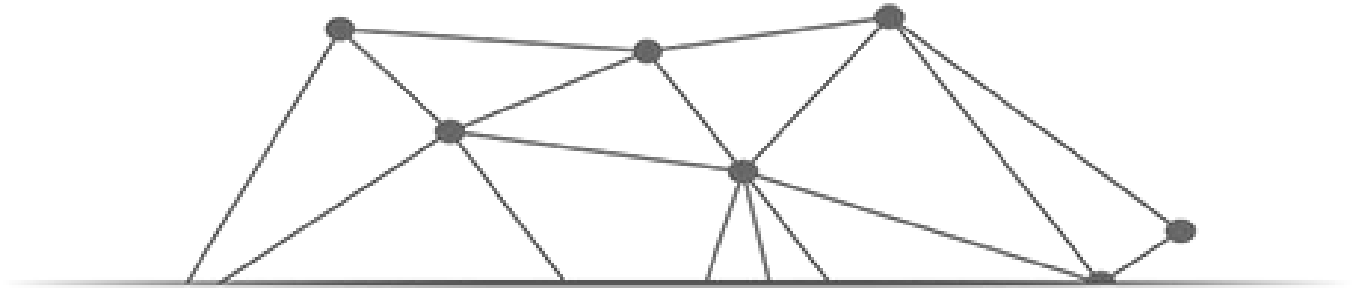
二叉树转换为森林

森林的数量等于根结点+右子树的数量(A->F->H)

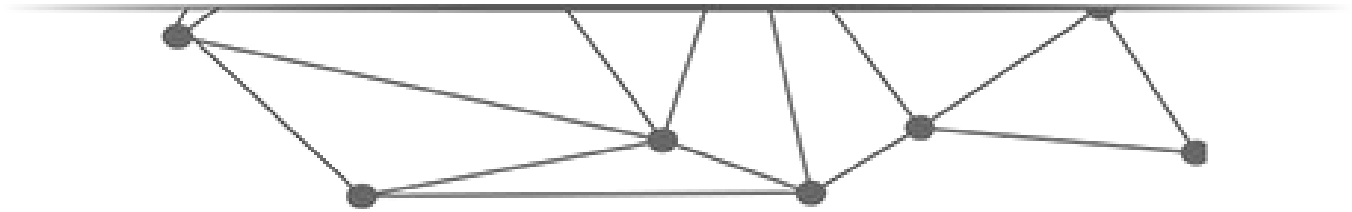
转换步骤：

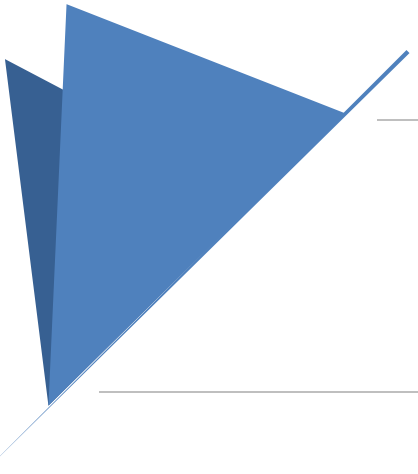
1. **整形：**将二叉树中所有结点的左孩子连线调整为垂直形态，右孩子调整为水平形态；
2. **连线：**对于任意水平线上的结点，将第一个结点的父节点作为该水平线上所有其他结点的父节点；
3. **去线：**删除所有水平连线，即删除原二叉树中各结点与其右孩子的连线；
4. **整形：**调整步骤3结束之后的树的形状，使其符合树的一般特征要求（父子、兄弟关系清晰）。





课堂互动 12.1





树和森林的遍历

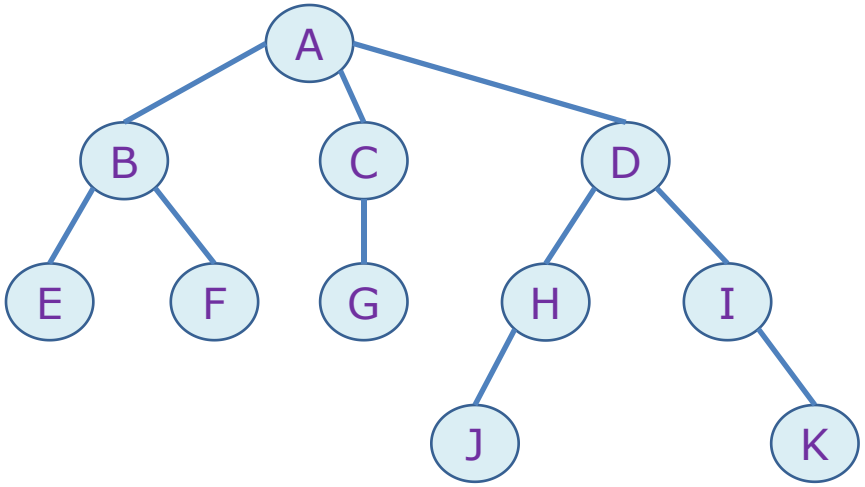
/ 树的遍历：先根遍历、后根遍历

/ 按深度遍历森林：先序遍历、中序遍历

/ 按宽度遍历森林：宽度遍历（层次遍历）

按深度方向遍历

树的先根遍历



先根遍历的结果：

A B E F C G D H J I K

基本规则：

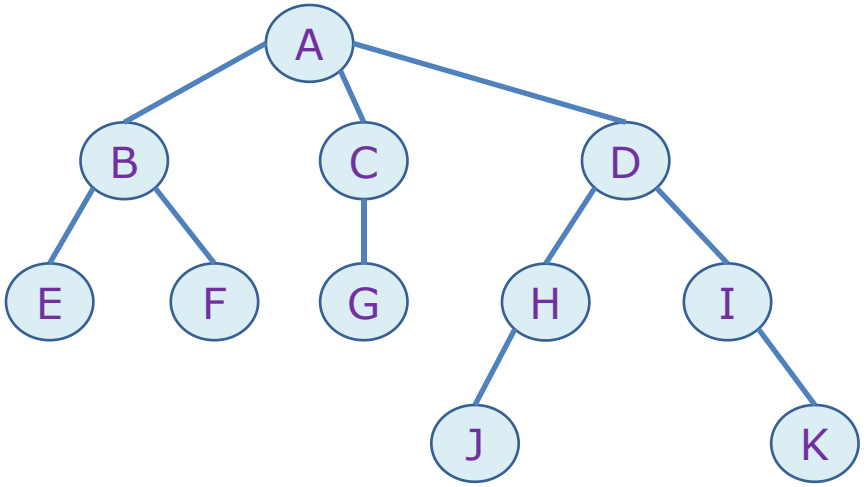
若树为空，则返回，遍历结束；否则：

1. 先访问根结点；
2. 按照从左到右的顺序，先根遍历根结点的每一棵子树（对照二叉树的先序遍历）。
3. 反复执行**步骤1-2**，直至遍历结束。

对树的先根遍历**等价于**对应二叉树的先序遍历

按深度方向遍历

树的后根遍历



后根遍历的结果：

E F B G C J H K I D A

基本规则：

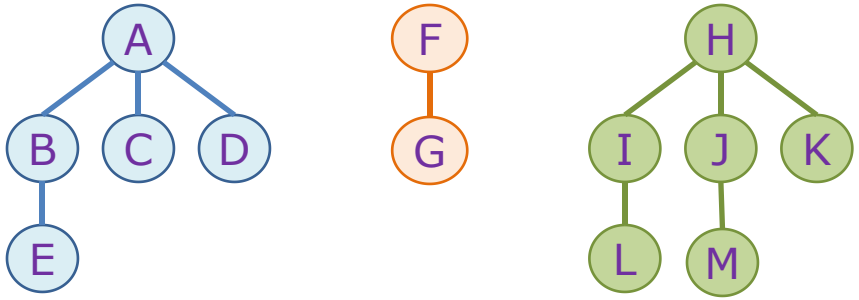
若树为空，则返回，遍历结束；否则：

1. 按照从左到右的顺序，后根遍历根结点的每一棵子树（对照二叉树的后序遍历）。
2. 先访问根结点；
3. 反复执行**步骤1-2**，直至遍历结束。

对树的后根遍历**等价于**对应二叉树的中序遍历

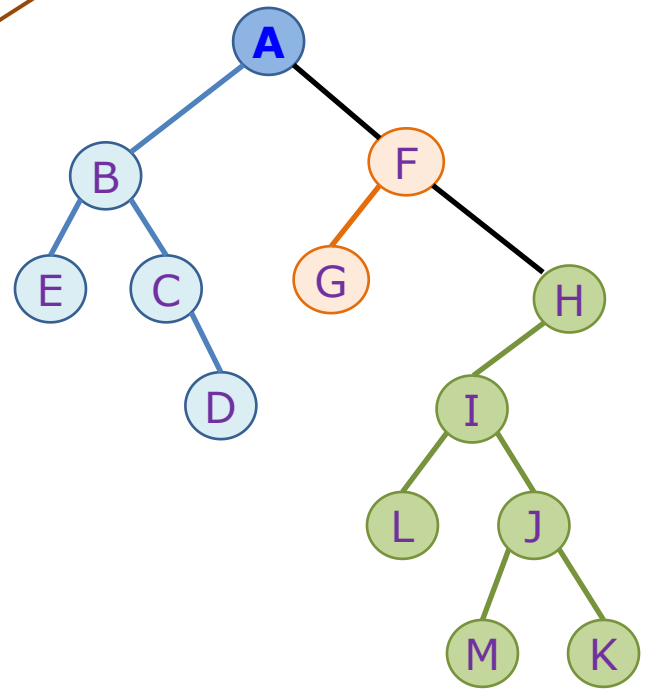
按深度方向遍历

森林的先序遍历



森林先序遍历的结果：A B E C D F G H I L J M K

与对应二叉树的先序遍历结果完全相同



基本规则：

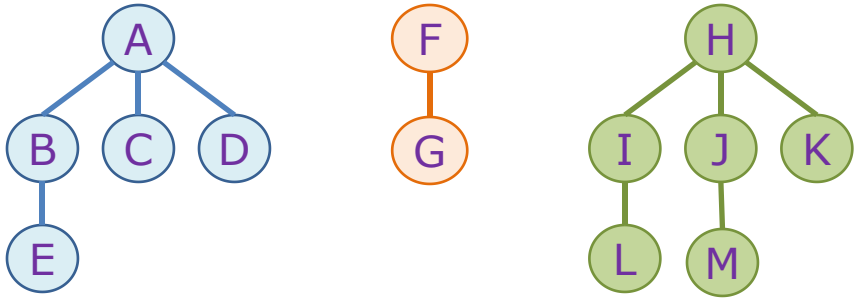
若森林为空，则返回，遍历结束；否则：

1. 访问第一棵树的根；
2. 先序遍历第一棵树的根结点的子树所构成的森林；
3. 先序遍历其余树构成的森林；
4. 对于每一棵子树的根结点所构成的子树，执行**步骤1-3**。

对森林的先序遍历**等于**对每棵树先序遍历的简单拼接

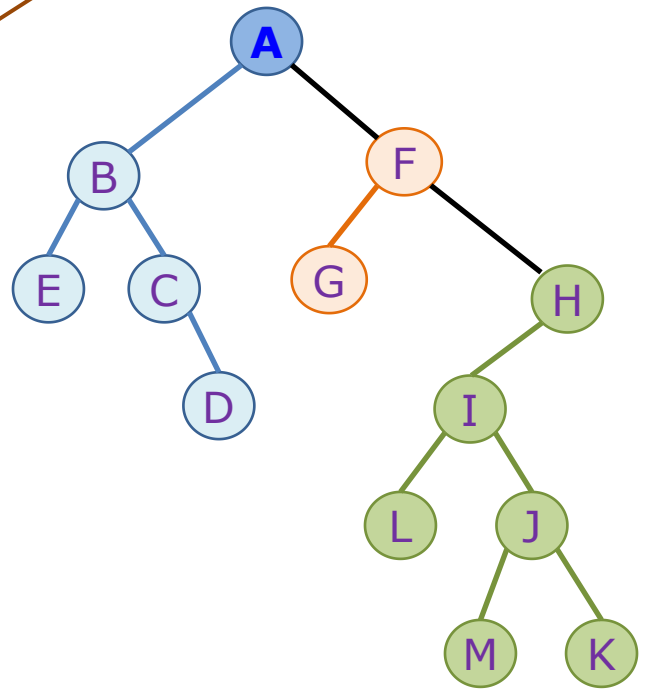
按深度方向遍历

森林的中序遍历



森林中序遍历的结果：E B C D A G F L I M J K H

与对应二叉树的先序遍历结果完全相同



基本规则：

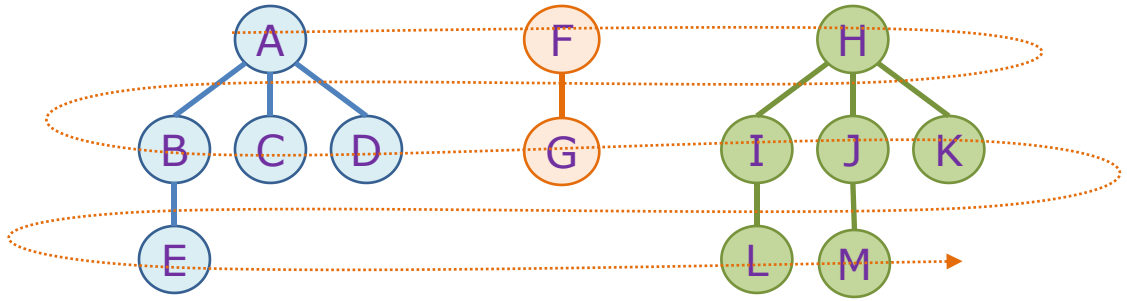
若森林为空，则返回，遍历结束；否则：

1. 中序遍历第一棵树的根结点的子树所构成的森林；
2. 访问第一棵树的根；
3. 中序遍历其余树构成的森林（对于每棵树使用后根遍历）；
4. 对于每一棵子树的根结点所构成的子树，执行**步骤1-3**。

对森林的中序遍历**等于**对每棵树后根遍历的简单拼接

按宽度方向遍历

森林的宽度（层次）遍历



森林先序遍历的结果：

A F H B C D G I J K E L M

基本规则：

若森林为空，则返回，遍历结束；否则：

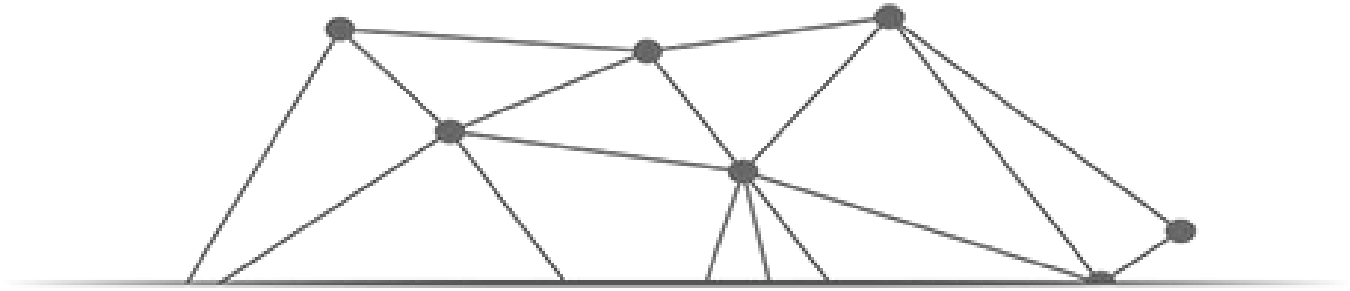
1. 按照从左到右的顺序，依次访问每棵树处于第一层的结点；
2. 按照同样的顺序，依次访问处于第二层的结点；
3. 依次类推，直至最下层的结点。

树和森林的遍历

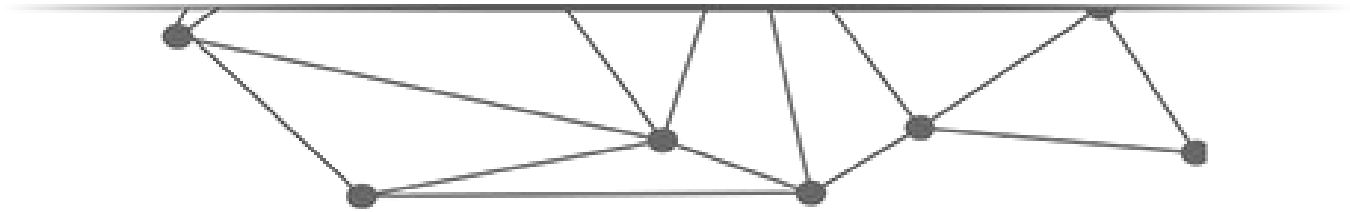
树、森林和二叉树遍历的对应关系

树	森林	二叉树
先根（序）遍历	先序（序）遍历	先序遍历
后根（序）遍历	中序（序）遍历	中序遍历
后序遍历	后序遍历	后序遍历

树和森林的后序遍历在逻辑上不够自然，因此不常用



课堂互动 12.2



第11讲 树的基本概念

小　节

- 树的存储结构有多种，例如：双亲表示法（顺序存储）、孩子表示法（二叉链表）、孩子兄弟表示法等。其中，孩子兄弟表示法是最常用的。任意一棵树都可以通过孩子兄弟表示法转换为对应的二叉树。
- 树和森林都能双向地转换成二叉树，并利用二叉树的操作解决一般树的问题。
- 根据结点访问次序的不同，树常见遍历方法包括先根遍历、后根遍历，其后根遍历与对应的二叉树的中序遍历一致。
- 根据结点访问次序的不同，森林常见遍历方法包括先序遍历、中序遍历
- 按从上到下的顺序，可以实现森林的层次遍历

欧老师的联系方式

读万卷书 行万里路 只为最好的修炼

QQ：14777591 (宇宙骑士)

Email：ouxinyu@alumni.hust.edu.cn

Website: http://ouxinyu.cn

Tel：18687840023

地址：安宁校区 诚远楼201

南院 智能应用研究院A306-2