

# 第5章 树和二叉树

## 知识要点

- (1) 掌握一般树的定义与基本术语；
- (2) 掌握二叉树的定义、性质及存储结构；
- (3) 熟练掌握二叉树的遍历及递归和非递归的遍历算法；
- (4) 进一步了解哈夫曼树、二叉查找树、平衡二叉树、堆与优先队列等二叉树的多种应用；
- (5) 熟悉树的存储结构及树和森林与二叉树之间的转换方法，了解树和森林的遍历方法。

薄钧戈

2021年4月19日

设按照从上到下、从左到右的顺序从1开始对完全二叉树进行顺序编号，则编号为 $i$ 结点的右孩子结点的编号为（ ）。

☒ A  $2i+1$

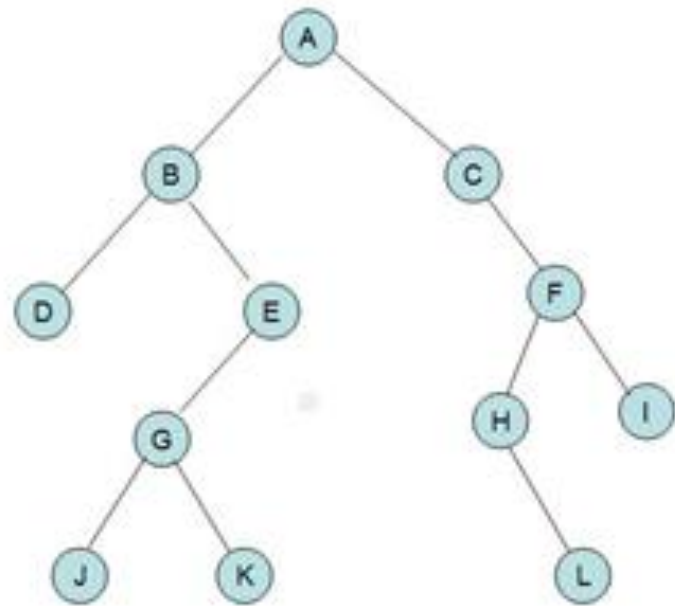
☐ B  $i$

☐ C  $i/2$

☐ D  $2i-1$

提交

对下面二叉树进行中序遍历，其序列是（ ）



- ☐ A ABDEGJKCFHLI
- ☒ B DBJGKEACHLFI
- ☐ C ABCDEFGHIJKL
- ☐ D DJKGEBLHIFCA

提交

设某棵二叉树中有2000个结点，则该二叉树的最小高度为（ ）

- ☐ A 8
- ☐ B 9
- ☐ C 10
- ☒ D 11

提交

关于哈夫曼树，下面描述错误的是（）

- ☐ A 哈夫曼树是二叉树
- ☐ B 生成该树是一种贪心策略
- ☒ C 生成的哈夫曼树必定唯一
- ☐ D 产生的哈夫曼编码有多种

提交

# 5.8 树与森林

## 5.8.1 树的存储结构

### 1. 双亲表示法

- 在树中，除了根结点没有双亲结点外，其余的结点都有唯一的一个双亲结点。在每个结点中附设一个指向其双亲结点的指针来唯一地表示一棵树，其结点结构图如图5-51所示。

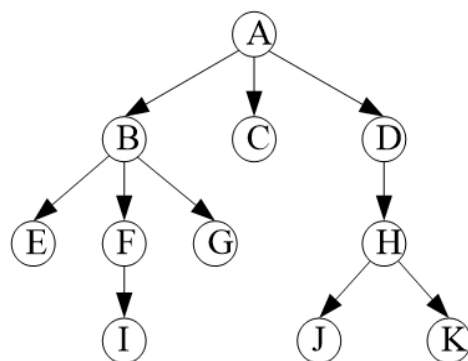


图 5-51 双亲表示法的结点结构图



## 5.8 树与森林

- 如图5-52(a)所示树 $T_9$ 的存储结构如图5-52(b)所示。为简明起见，将父指针表示为双亲结点在数组中位置的下标，注意根结点A无双亲，设其parent的值为-1。



(a) 树  $T_9$

0	A	-1
1	B	0
2	C	0
3	D	0
4	E	1
5	F	1
6	G	1
7	H	3
8	I	5
9	J	7
10	K	7

(b) 树的双亲表示法的存储结构

图 5-52 树的双亲表示法的存储结构

双亲表示法是实现树的最简单方法，使用这种方法在寻找树中某一结点的双亲时只需要 $O(1)$ 时间



# 5.8 树与森林

## 2. 孩子表示法

- 树中的每个结点可能有多个孩子，所以可以对树中的每个结点用一个包含结点本身数据的数据域和多个指向该结点孩子的指针域来表示一棵树，各指针域反映了树中结点与结点之间的关系。其结点结构图如图5-53所示，其中 $n$ 为树的度。



图 5-53 孩子表示法的结点结构图





## 5.8 树与森林

➡ 图5-54为图5-52(a)所示的树的孩子表示法的结构图。

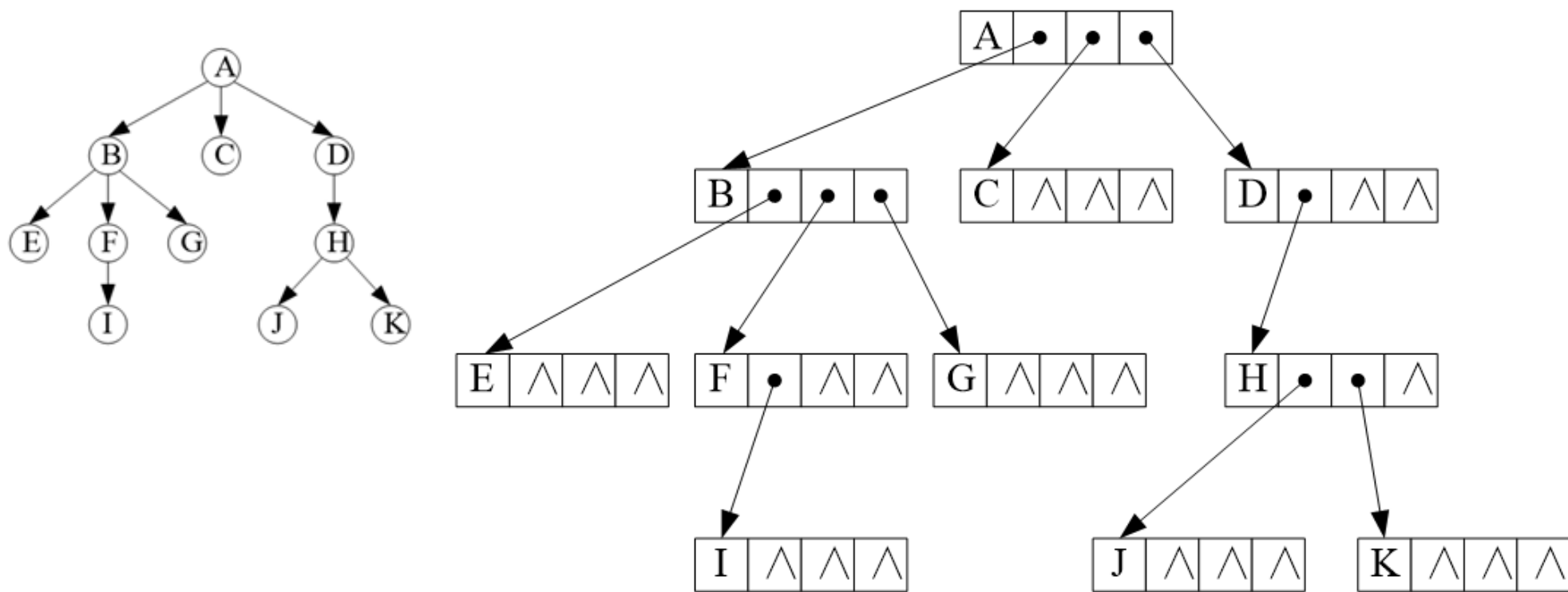


图 5-54 树的孩子表示法的存储结构图



## 5.8 树与森林

► 比较好的方法是将每个结点的孩子结点按照从左到右的顺序以**单链表**的形式存储起来，每个结点包含一个结点数据域以及一个指向子结点链表的指针。

► 图5-55为图5-52(a)所示的树的孩子表示法的链表存储结构。

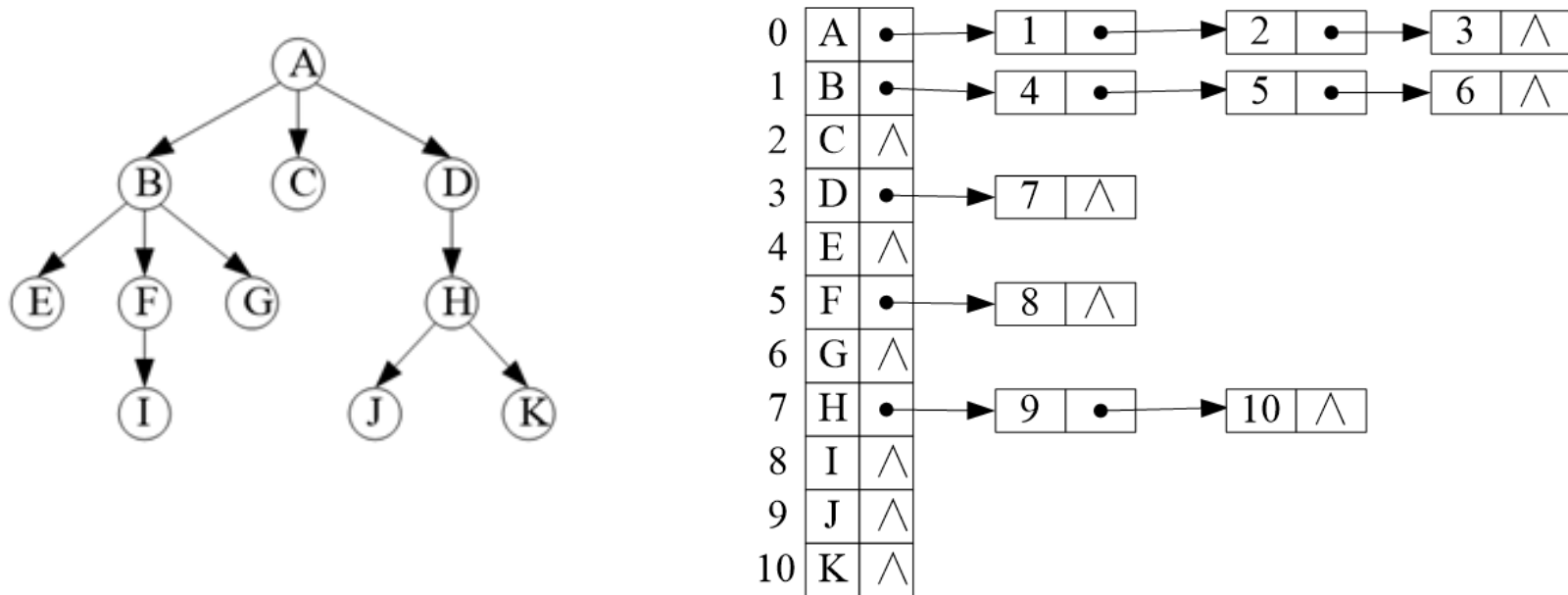


图 5-55 树的孩子表示法的链表存储结构



## 5.8 树与森林

### 3. 左孩子/右兄弟表示法

- 将每个结点用一个包含该结点的数据域、一个指向该结点最左孩子结点的指针域和右兄弟结点的指针域来表示，其结点结构图如图5-57所示。

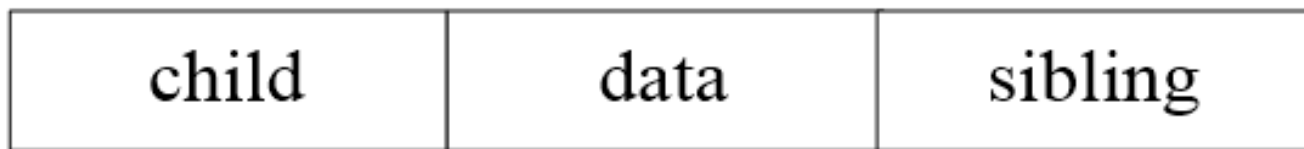


图 5-57 左孩子/右兄弟表示法结点结构图



## 5.8 树与森林

► 图5-58为图5-52(a)所示的树的左孩子/右兄弟表示法的链表存储结构。

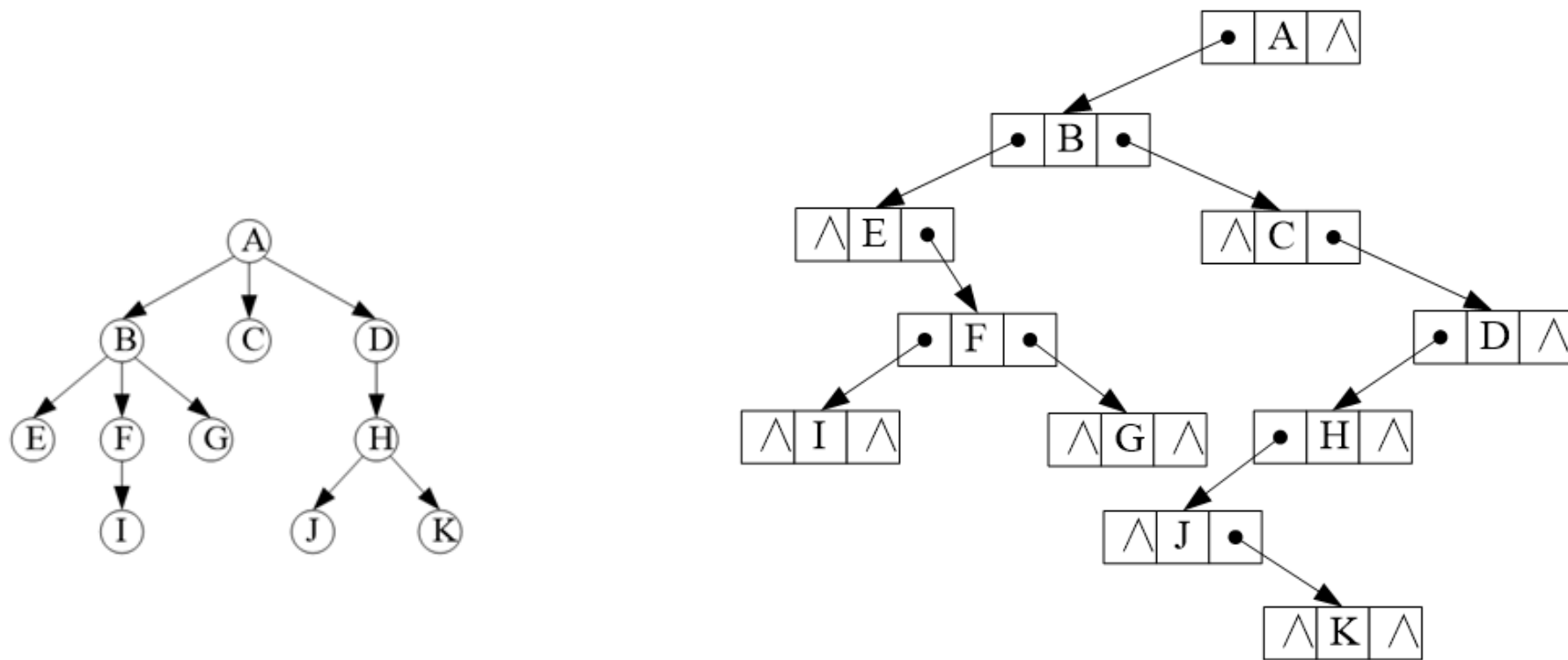


图 5-58 树的左孩子/右兄弟表示法的链式存储结构



## 5.8.2 树、森林与二叉树的转换

### 1. 树转换为二叉树

► 转换方法如下：

- 在所有兄弟结点之间加一条线；
- 对树中的每个结点，只保留它与最左边孩子的连线，删除该结点与其他孩子的连线；
- 以树根为轴心，对结点进行旋转处理即可得到相应的二叉树。



## 5.8.2 树、森林与二叉树的转换

► 如图5-59所示为将图5-59(a)的一般树转换为二叉树的过程。

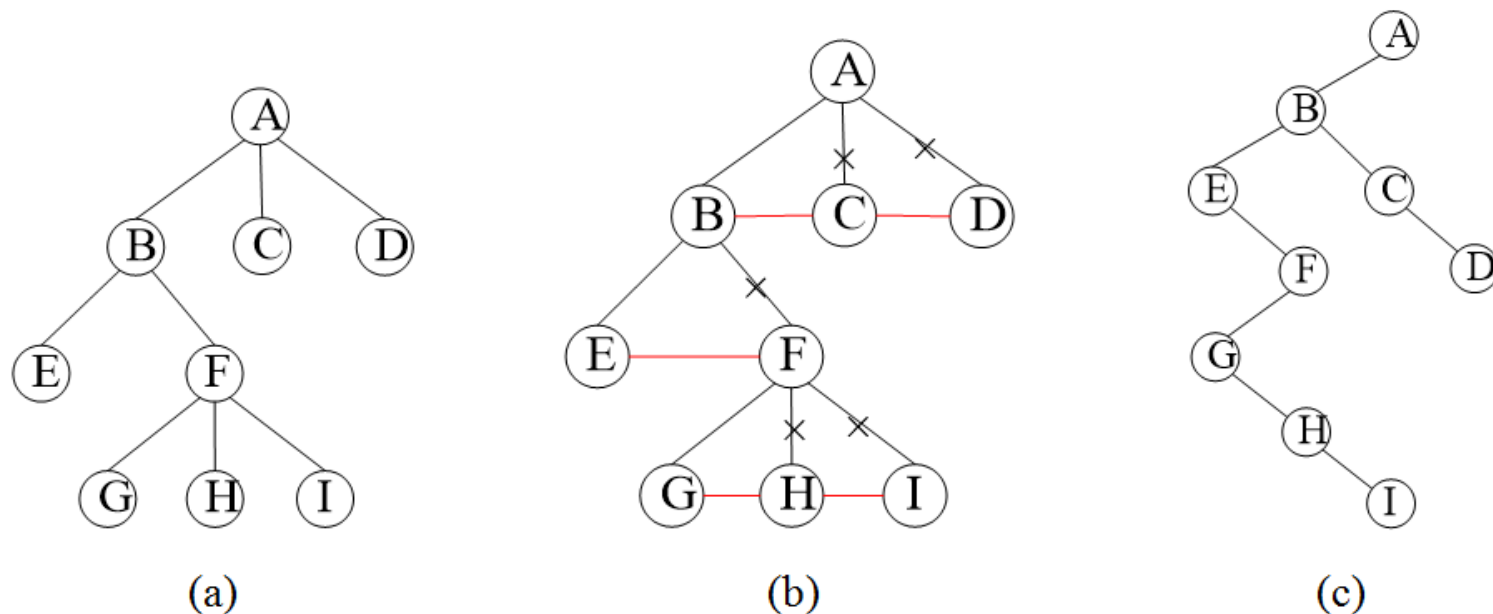


图 5-59 一般树转换为二叉树



## 5.8.2 树、森林与二叉树的转换

### 2. 森林转换为二叉树

转换方法如下：

- 将森林中的每棵树转换为二叉树；
- 从第一棵二叉树开始，依次将森林中的后一棵二叉树的根结点作为前一棵二叉树根结点的右孩子连接在一起；
- 以第一棵二叉树的树根为轴心，对结点进行旋转处理即可得到相应的二叉树。



## 5.8.2 树、森林与二叉树的转换

► 如图5-60所示为将图5-60(a)的森林转换为二叉树的过程。

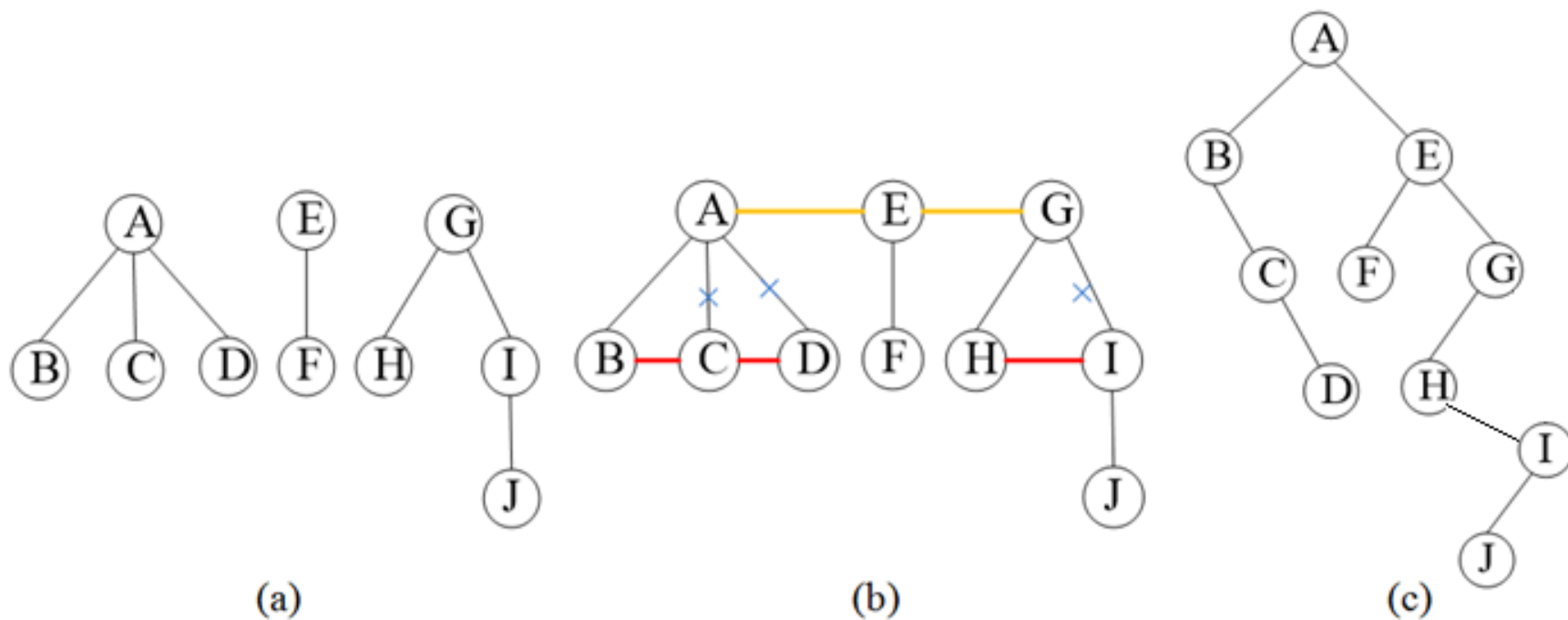


图 5-60 森林转换为二叉树





## 5.8.2 树、森林与二叉树的转换

### 3. 二叉树转换为树或森林

转换步骤如下：

- (1) 若二叉树的树根结点存在右子树，先将其与右子树的连线删除，得到分离的二叉树；
- (2) 将分离后的二叉树按照以下步骤转换为树：若某结点的左孩子存在，将左孩子的右孩子，右孩子的右孩子……都作为该结点的孩子结点用线连接起来，然后删除掉原二叉树中所有结点与其右孩子的连线即可；
- (3) 整理(2)得到的树即为分离的二叉树转换后的树；
- (4) 若剩余二叉树非空，则转到(1)完成森林其余树转换。



## 5.8.2 树、森林与二叉树的转换

► 如图5-61为二叉树转换为森林的过程。

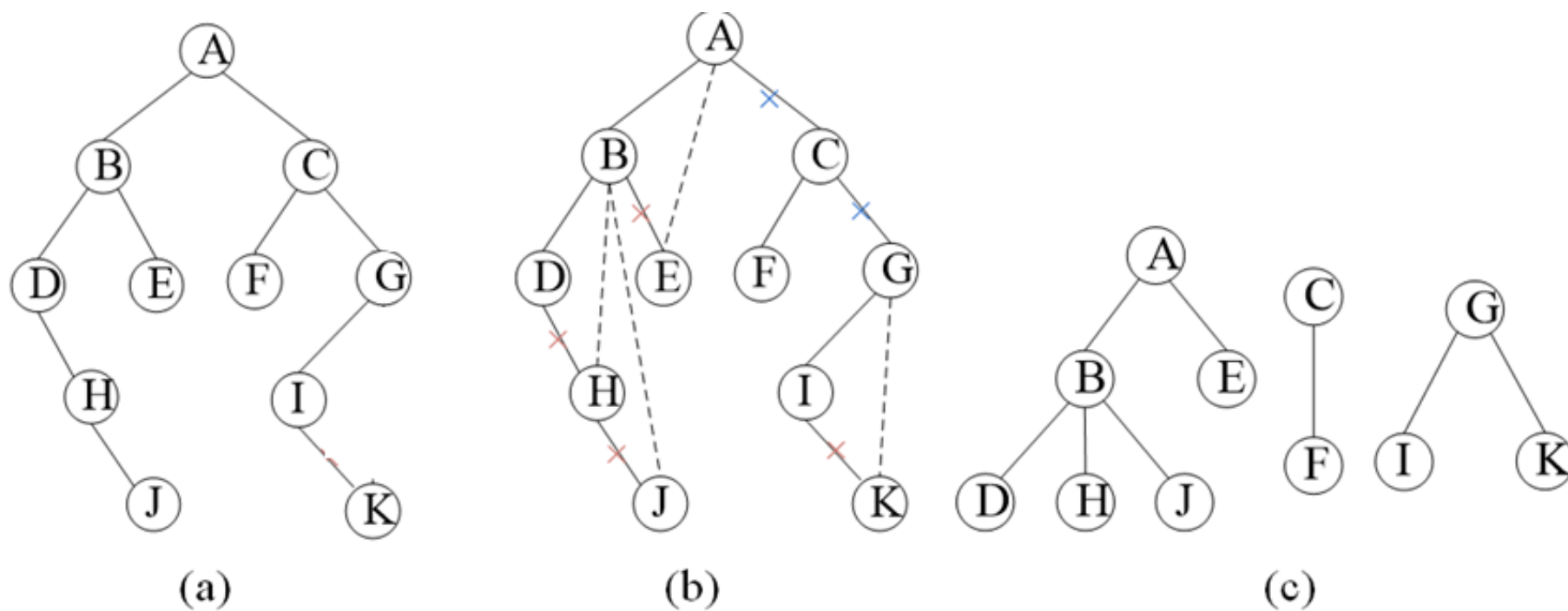


图 5-61 二叉树转换为森林



## 5.8.3 树与森林的遍历

- 树有两种遍历方法：先根遍历和后根遍历。设树 $T$ 如图5-62所示，结点 $R$ 是根，根的子树从左到右依次为 $T_1, T_2, \dots, T_k$ 。

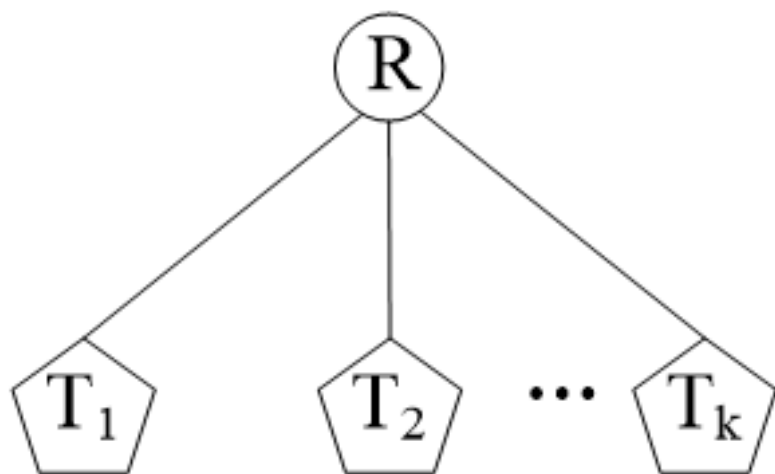


图 5-62 树  $T$



## 5.8.3 树与森林的遍历

### (1) 先根（序）遍历树

➡ 若树 $T$ 非空，则：

➡ 访问根结点 $R$ ；

➡ 依次先根遍历根 $R$ 的各子树 $T_1, T_2, \dots, T_k$ 。

### (2) 后根（序）遍历树

➡ 若树 $T$ 非空，则：

➡ 依次后根遍历根 $T$ 的各子树 $T_1, T_2, \dots, T_k$ ；

➡ 访问根结点 $R$ 。

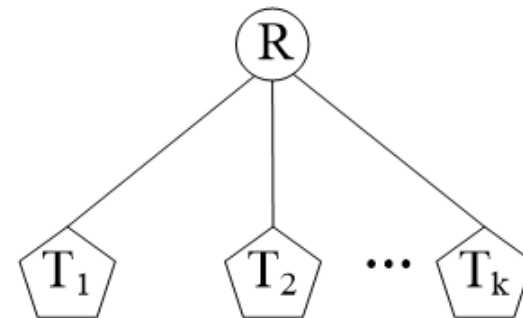


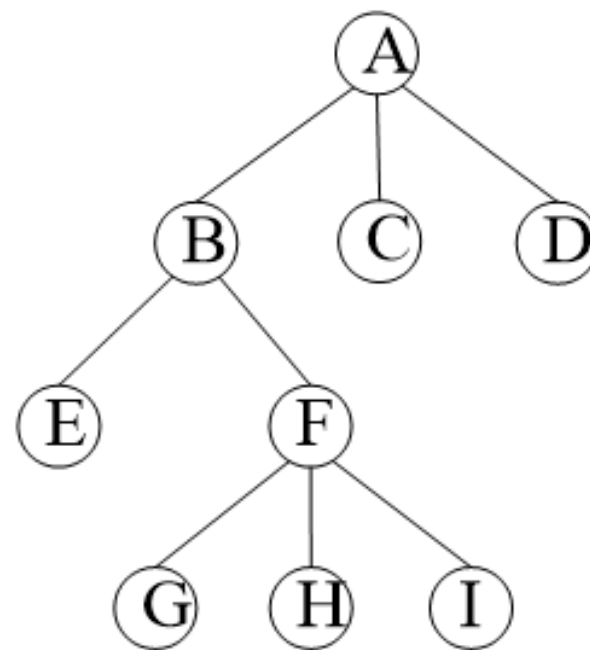
图 5-62 树  $T$



## 5.8.3 树与森林的遍历

对图5-59(a)所示的树进行先根遍历和后根遍历，

- 先根遍历序列：ABEFGHICD
- 后根遍历序列：EGHIFBCDA



(a)



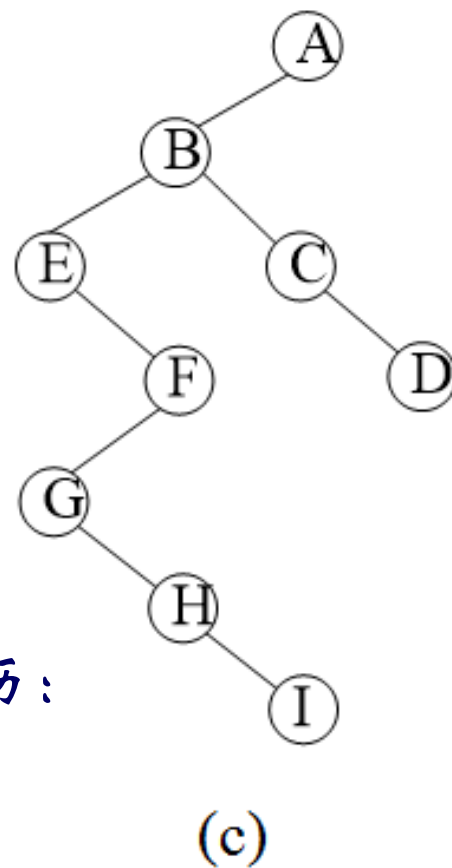
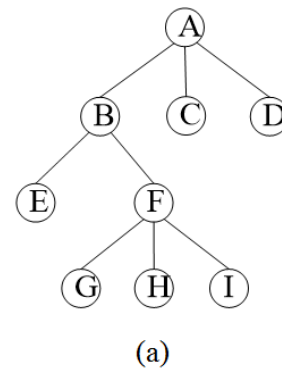
## 5.8.3 树与森林的遍历

■ 对图5-59(a)所示的树转换后对应的二叉树如图5-59(c)进行,

■ 先序遍历: ABEFGHICD

■ 中序遍历: EGHIFBCDA

■ 后序遍历: IHGFEDCBA



对图5-59(a)所示的树进行先根遍历和后根遍历:

先根遍历序列: ABEFGHICD

后根遍历序列: EGHIFBCDA



## 5.8.3 树与森林的遍历

森林的遍历定义如下：

### (1) 先根（序）遍历森林

➡ 若森林非空，则：

- ➡ 访问森林中第一棵树的根结点；
- ➡ 先根遍历第一棵树中的子树森林；
- ➡ 先根遍历除去第一棵树之后剩余的树构成的森林。

### (2) 中根（序）遍历森林

- ➡ 中根遍历森林中第一棵树的根结点的子树森林；
- ➡ 访问森林中第一棵树的根结点；
- ➡ 中根遍历除去第一棵树之后剩余的树构成的森林。



## 5.8.3 树与森林的遍历

### ■ 森林:

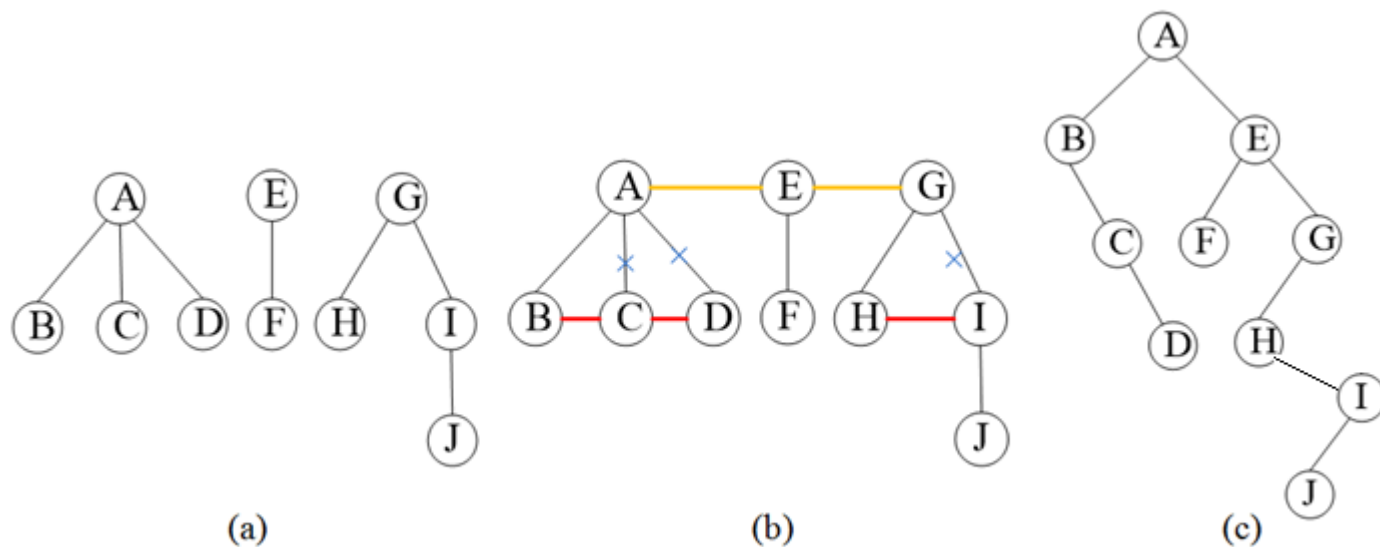
■ 先根遍历: ABCDEFGIJ

■ 中根遍历: BCDAFEHJIG

### ■ 森林转换后对应的二叉树:

■ 先序遍历: ABCDEFGHIJ

■ 中序遍历: BCDAFEHJIG





END