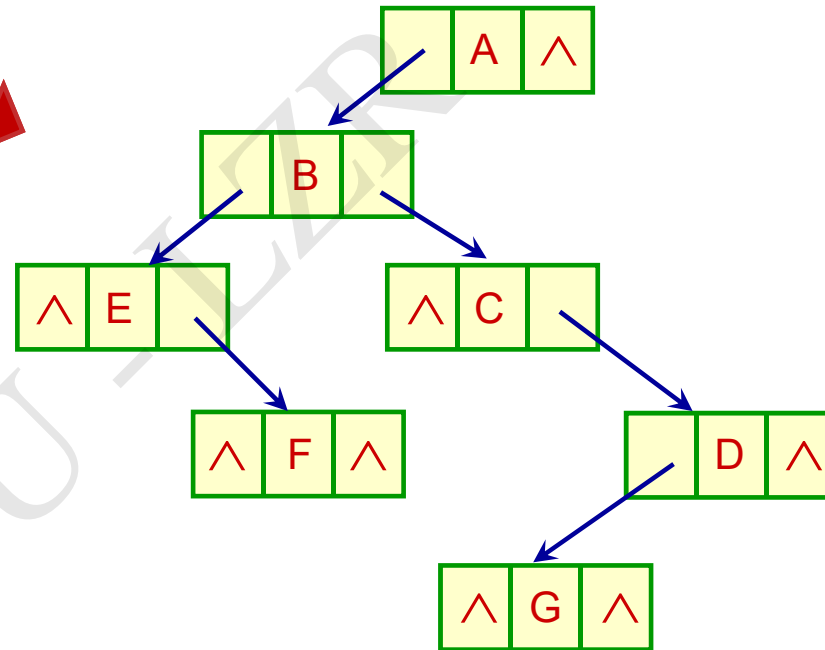
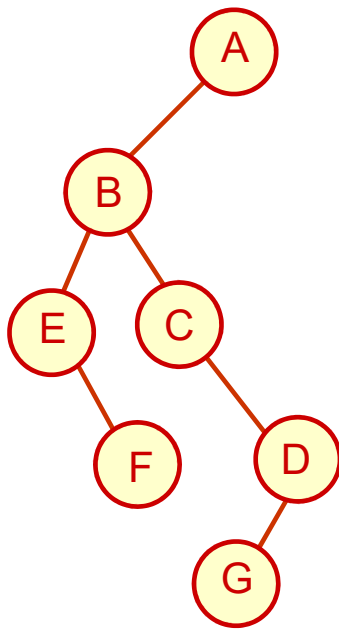
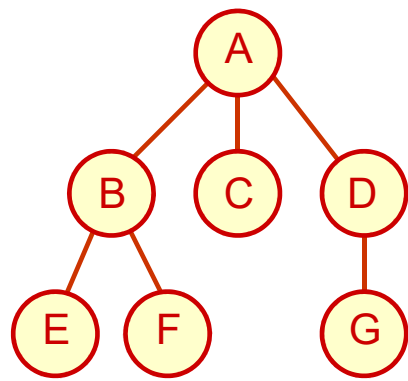


6.4.2 森林与二叉树的转换

- 由于二叉树和树都可用二叉链表作为存储结构，则以二叉链表作为媒介可导出树与二叉树之间的一个对应关系。
- 也就是说，给定一棵树，可以找到惟一的一棵二叉树与之对应，从物理结构来看，它们的二叉链表是相同的，只是解释不同而已。



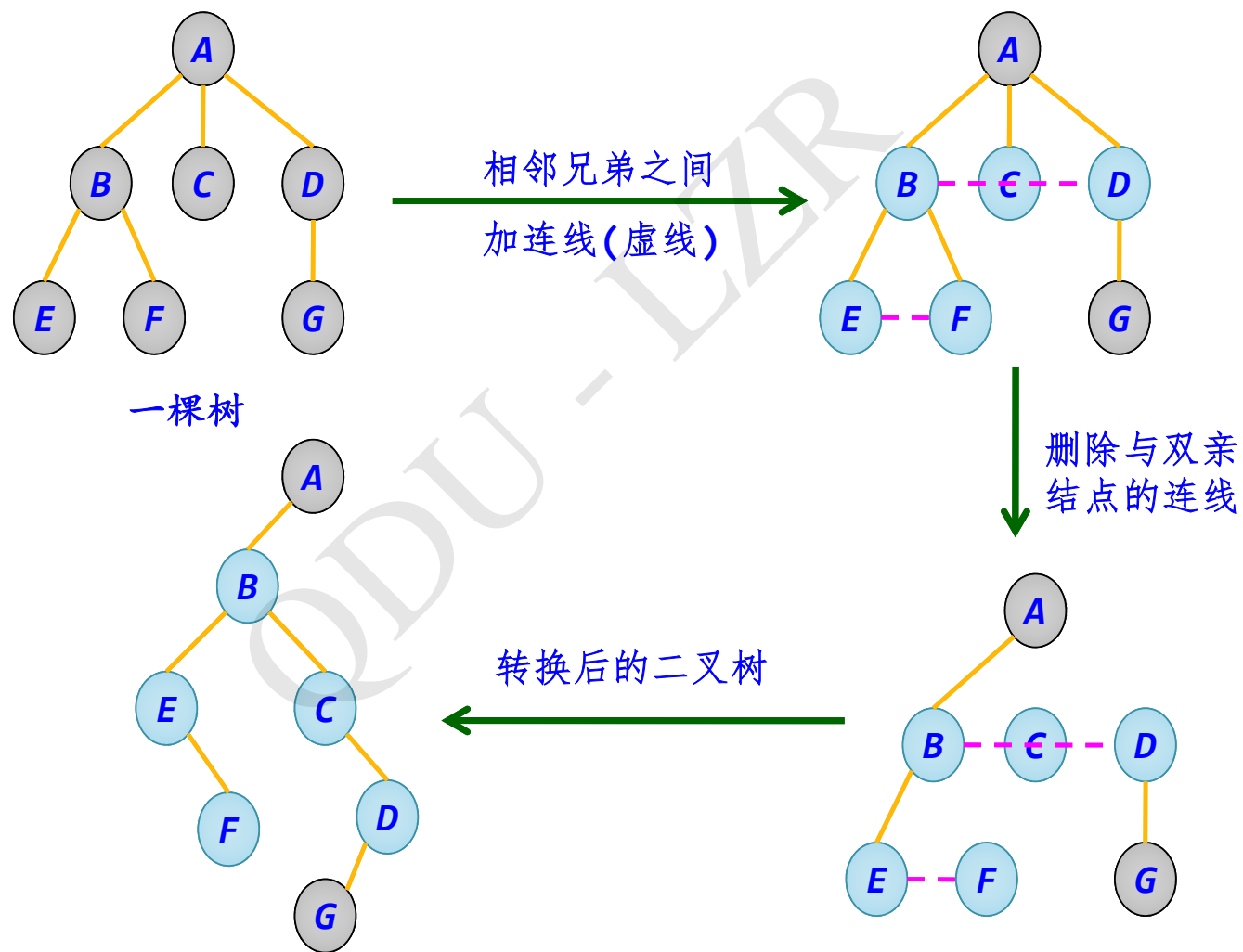
6.4.2 森林与二叉树的转换

将一棵树转换成二叉树的过程如下：

- 树中所有相邻兄弟之间加一条连线；
- 对树中的每个结点，只保留它与第一个孩子结点之间的连线，删除它与其他孩子结点之间的连线；
- 以树的根结点为轴心，将整棵树顺时针转动45度，使之结构层次分明。

【示例-1】 将下图所示的树转换成二叉树。

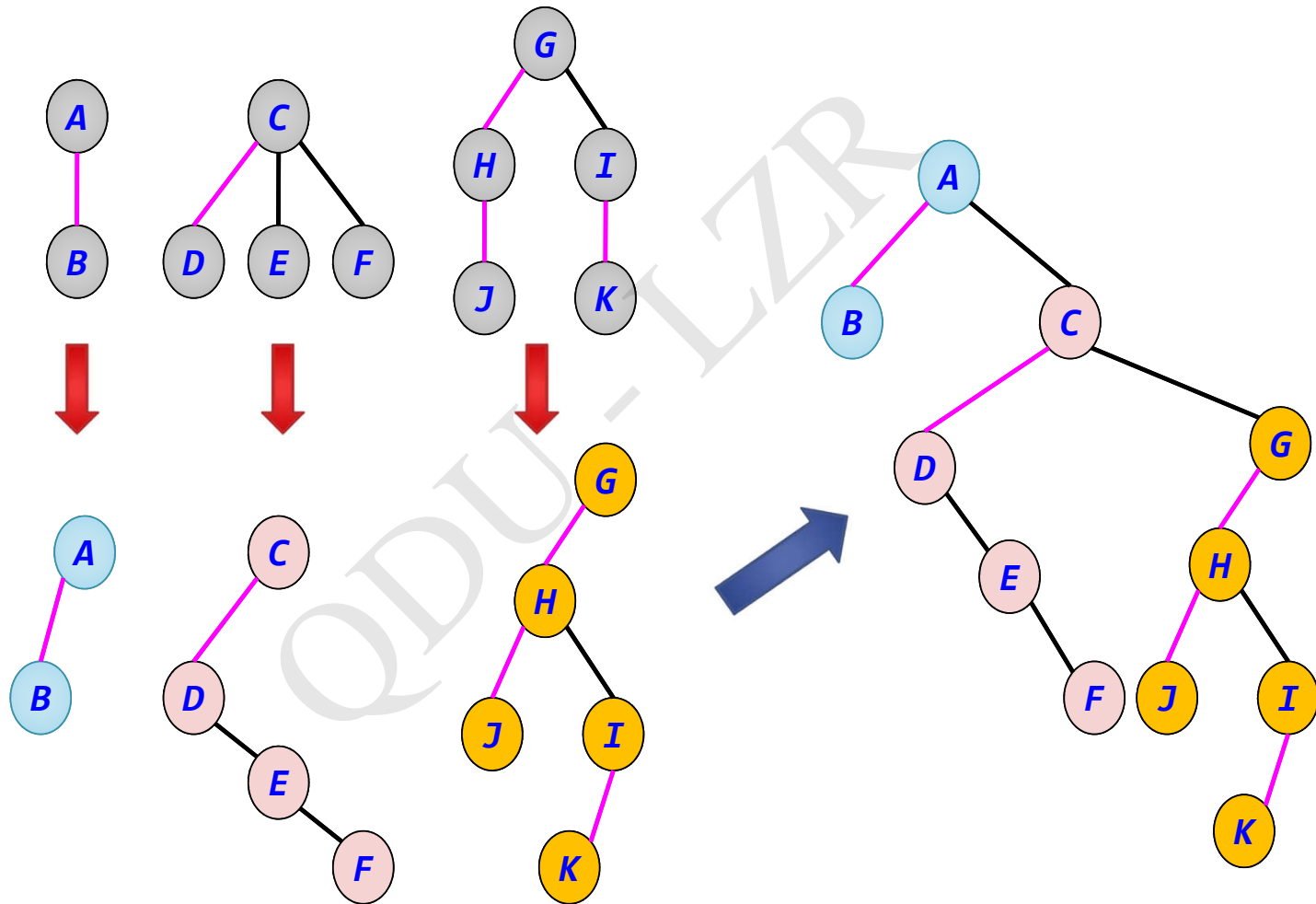
解：转换的过程：



当要转换为二叉树的森林由两棵或以上树构成时，将这样的森林转换为二叉树的过程如下：

- 将森林中的每棵树转换成相应的二叉树。
- 第一棵二叉树不动，从第二棵二叉树开始，依次把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结点，当所有二叉树连在一起后，此时所得到的二叉树就是由森林转换得到的二叉树。

【示例-2】 将下图所示的森林转换成二叉树。

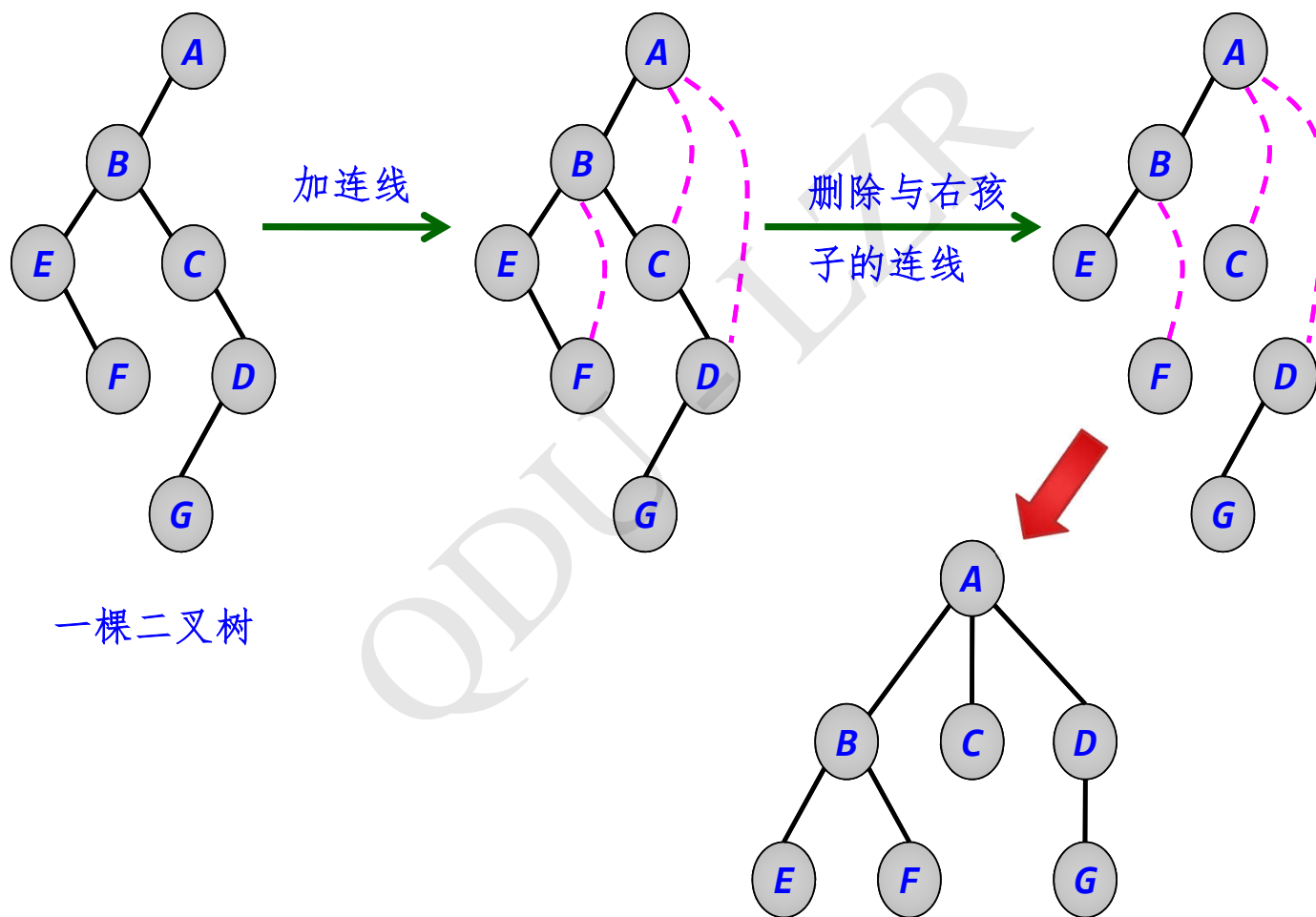


6.4.3 二叉树还原为树/森林

当一棵二叉树是由一棵树转换而来的，则该二叉树还原为树的过程如下：

- 若某结点是其双亲的左孩子，则把该结点的右孩子、右孩子的右孩子、…、都与该结点的双亲结点用连线连起来。
- 删除原二叉树中所有双亲结点与右孩子结点之间的连线。
- 整理由①、②两步所得到的树，使之结构层次分明。

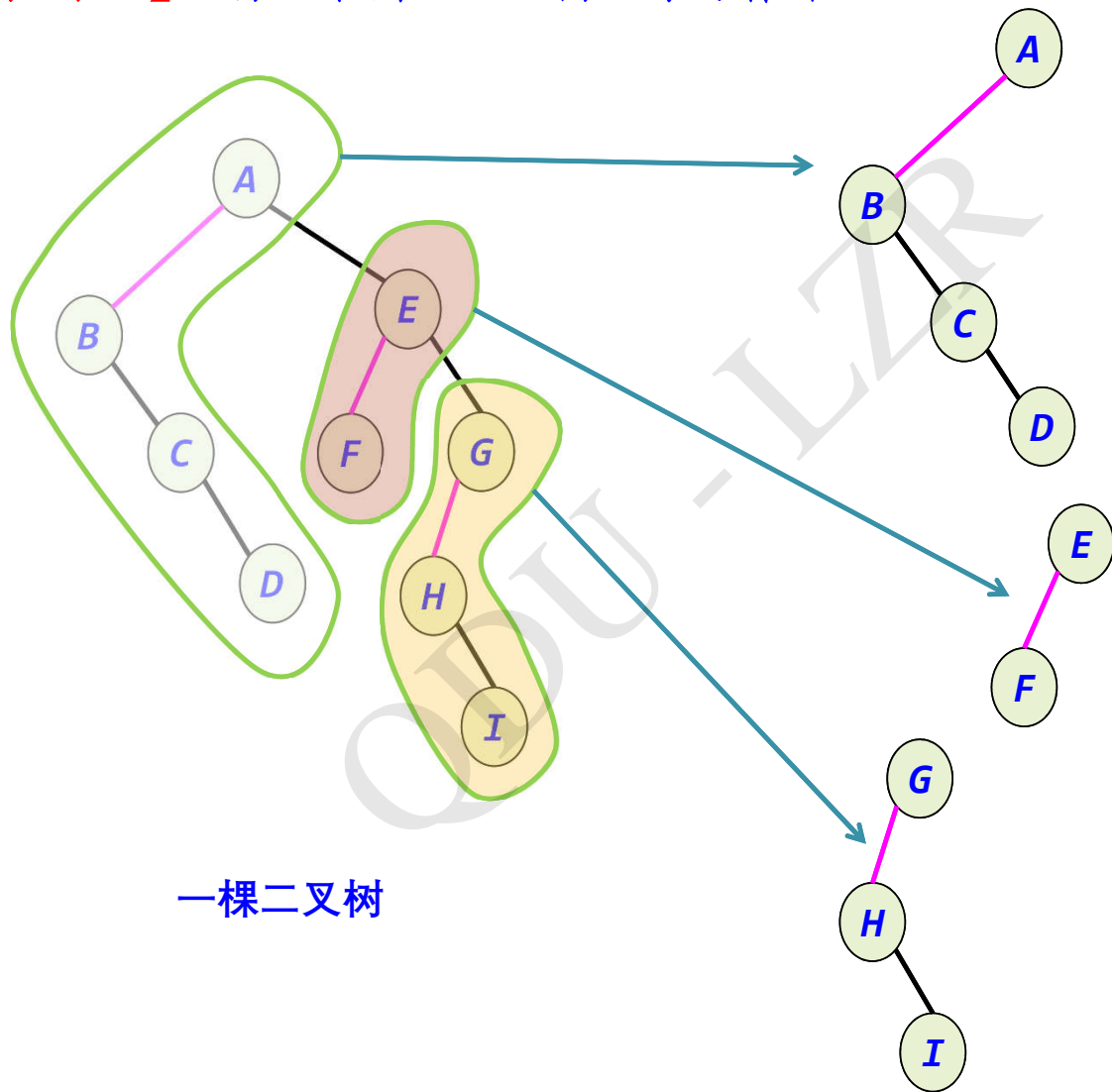
【示例-3】 将下图的一棵二叉树还原为树。

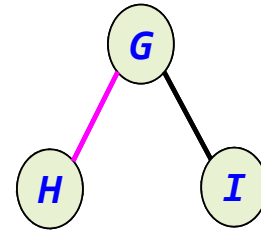
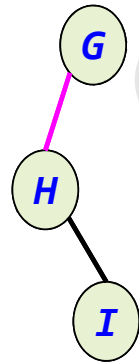
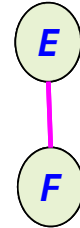
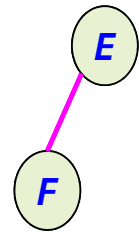
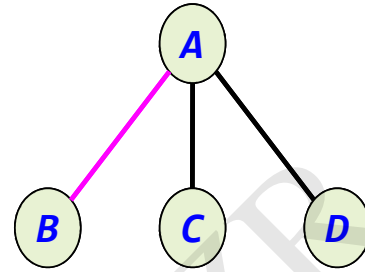
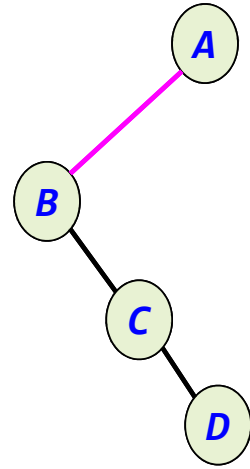


当一棵二叉树是由 m 棵树构成的森林转换而来的，该二叉树的根结点一定有 $m-1$ 个右下孩子，则该二叉树还原为森林的过程如下：

- 抹掉二叉树根结点右链上所有结点之间的“双亲—右孩子”关系，将其分成若干个以右链上的结点为根结点的二叉树，设这些二叉树为 bt_1 、 bt_2 、...、 bt_m 。
- 分别将 bt_1 、 bt_2 、...、 bt_m 二叉树各自还原成一棵树。

【示例-4】 将如下图的二叉树还原为森林。





6.4.3 树的遍历

树的遍历运算是指按某种方式访问树中的每一个结点且每一个结点只被访问一次。

有以下两种遍历方法：

- 先根遍历
- 后根遍历

注意：先根和后根遍历算法都是递归的。

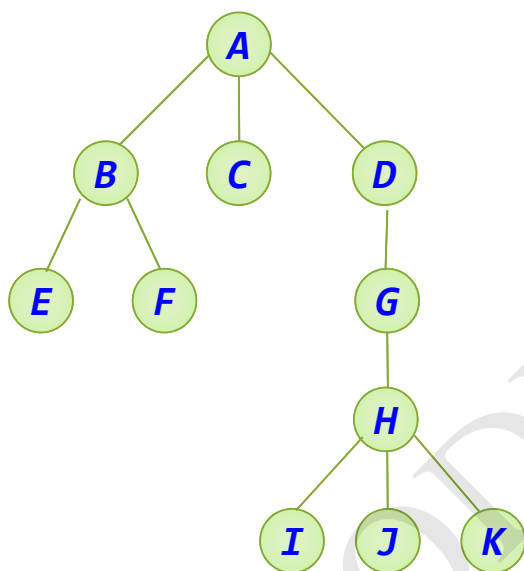
先根遍历:

若树不空, 则先访问根结点, 然后依次先根遍历各棵子树。

后根遍历:

若树不空, 则先依次后根遍历各棵子树, 然后访问根结点。

【示例-5】 遍历下图所示的树。

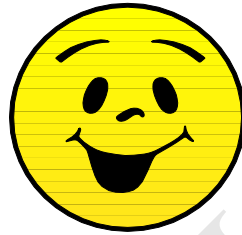


先根遍历的顶点访问次序：

A B E F C D G H I J K

后根遍历的顶点访问次序：

E F B C I J K H G D A



— END —