

# 数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program



## 树、森林及二叉树的相互转换

- 从一个屌丝逆袭高富帅的小故事说起。
- 在这一章节开始的时候我们是从一棵普通的树开始介绍，在满足树的条件下可以是任意形状，一个结点可以有任意多个孩子，这样对树的处理显然要复杂很多。
- 所以我们研究出了一些条条框框的限定，如：二叉树，完全二叉树，满二叉树等。
- 那么这时候你就会想，如果所有的树都像二叉树一样方便处理就好了。



## 普通树转换为二叉树

- 步骤如下:
  - 加线, 在所有兄弟结点之间加一条连线。
  - 去线, 对树中每个结点, 只保留它与第一孩子结点的连线, 删除它与其他孩子结点之间的连线。
  - 层次调整, 以树的根结点为轴心, 将整棵树顺时针旋转一定的角度, 使之结构层次分明。



## 森林转换为二叉树

- 步骤如下:
  - 把每棵树转换为二叉树。
  - 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的根结点的右孩子, 用线连接起来。



## 二叉树转换为树、森林

- 二叉树转换为普通树是刚才的逆过程，步骤也就是反过来做而已。
- 判断一棵二叉树能够转换成一棵树还是森林，标准很简单，那就是只要看这棵二叉树的根结点有没有右孩子，有的话就是森林，没有的话就是一棵树。



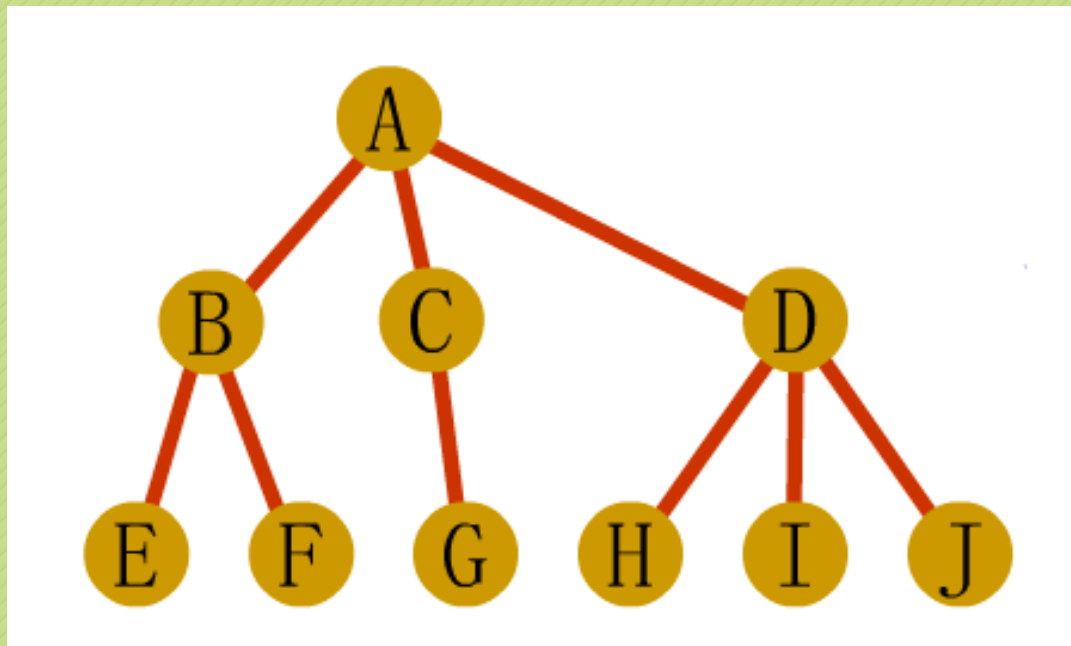


## 树与森林的遍历

- 树的遍历分为两种方式：一种是先根遍历，另一种是后根遍历。
- 先根遍历：先访问树的根结点，然后再依次先根遍历根的每棵子树。
- 后根遍历：先依次遍历每棵子树，然后再访问根结点。
- No pic you say a J8!



## 树与森林的遍历



- 先根遍历结果: ABEFCGDHIJ
- 后根遍历结果: EFBGCHIJDA



## 树与森林的遍历

- 森林的遍历也分为前序遍历和后序遍历，其实就是按照树的先根遍历和后根遍历依次访问森林的每一棵树。
- 我们的惊人发现：树、森林的前根（序）遍历和二叉树的前序遍历结果相同，树、森林的后根（序）遍历和二叉树的中序遍历结果相同！
- 这其实也就证实我们视频开头讲解的那个例子，我们找到了对树和森林遍历这种复杂问题的简单解决方案！

