



树

学习目标

树的定义
树的基本术语
树的抽象数据类型定义



树的提出

文件系统

例 1 操作系统的文件目录结构

20计科数据结构

data structure

数据结构

材料

练习、复习

王红梅

英文教材

程序测试

第1周

第2周

第3周

第4周

第5-6周

第7周

第8周

第9周

第10周

第11周

第12周

5树与二叉树.pptx

8-4选择类排序.pptx

8-5归并排序、基数排序及排序总结.pptx

12周学习要求 (排序2) .pptx

12周学习要求 (排序2) s.pptx

许多问题抽象出的数据模型具有层次关系

表达式树

例 2 编译器的语法检查——二叉表示树

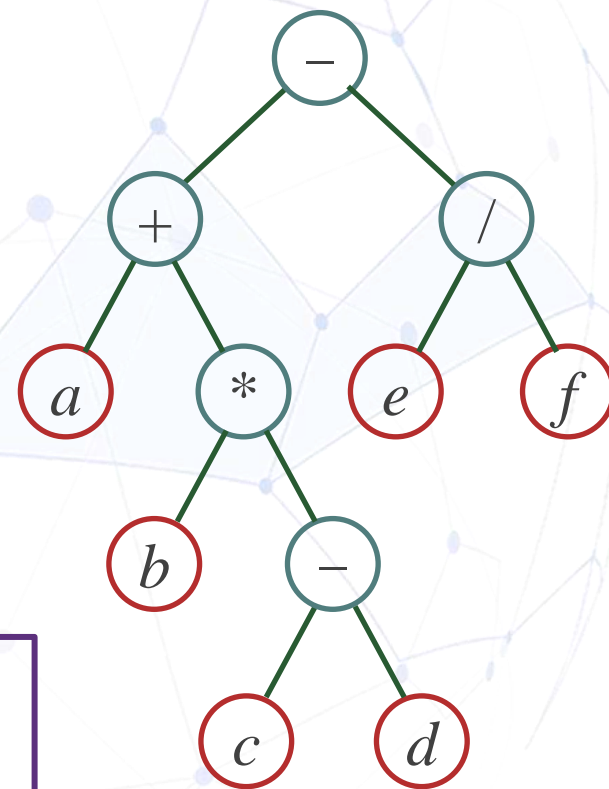
一个算术表达式可以用二叉表示树来表示，通过二叉表示树可以判断算术表达式是否存在语法错误。二叉表示树具有以下两个特点：

(1) 叶子结点是操作数； (2) 分支结点是运算符。

表达式： $a + b * (c - d) - e / f$

进行后序遍历转换为逆波兰式：

$a b c d - * + e f / -$



波兰式： 1929年波兰数学家提出一种前缀表示法，可以不用考虑运算符的优先级而直接计算表达式的值

随处可见的树结构

例 3 计算机系统中随处可见的树结构——菜单

查看(V)

排序方式(O)

刷新(E)

粘贴(P)

粘贴快捷方式(S)

撤销 重命名(U)

360桌面助手

图形属性.....

图形选项

新建(W)

屏幕分辨率(C)

小工具(G)

大图标的(R)

中等图标(M)

小图标(N)

自动排列图标(A)

将图标与网格对齐(I)

显示桌面图标(D)

显示桌面小工具(G)

Ctrl+Z

5-1引言.pptx - Microsoft

开始 插入 设计 动画 幻灯片放映 审阅 视图 加载项

从头开始 从当前幻灯片开始 自定义幻灯片放映

设置 隐藏 幻灯片放映 幻灯片

录制旁白 排练计时 使用排练计时

分辨率: 使用当前分辨率

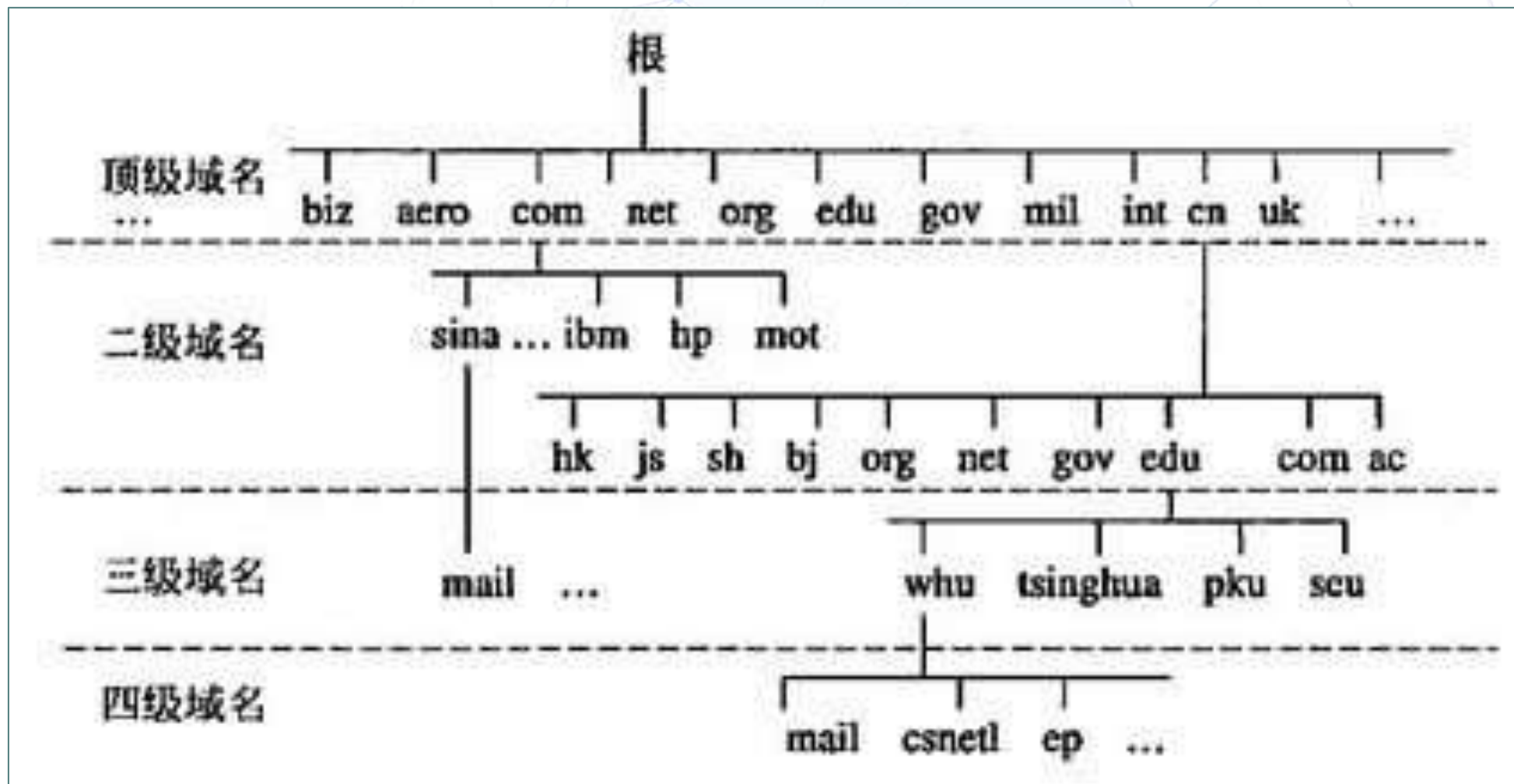
演示文稿显示位置:

使用演示者视图

监视器

随处可见的树结构

例 3 计算机系统上随处可见的树结构——internet的名字空间



随处可见的树结构

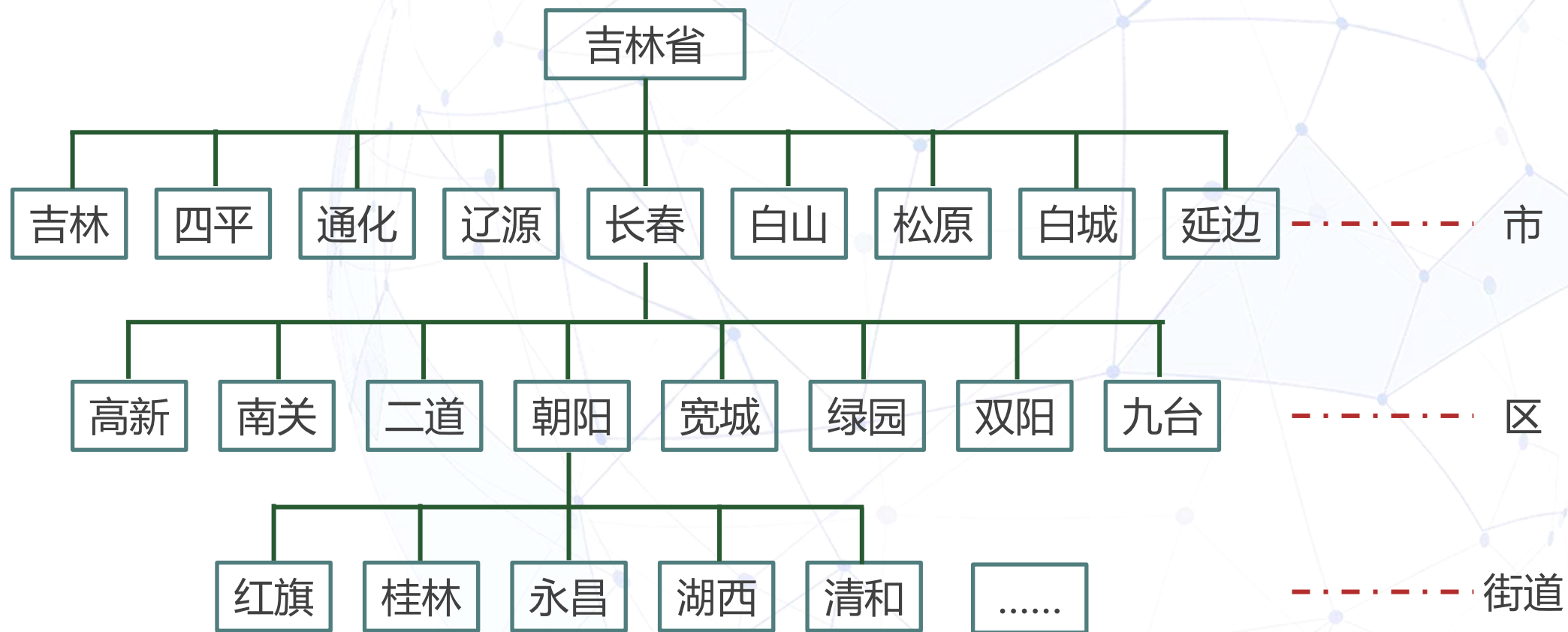
例 4 日常生活中随处可见的树结构——目录、家谱

目 录

第 1 章 绪 论.....	1
1.1 问题求解与程序设计	2
1.1.1 程序设计的一般过程.....	2
1.1.2 数据结构在程序设计中的作用.....	5
1.1.3 算法在程序设计中的作用.....	6
1.1.4 本书讨论的主要内容.....	7
1.2 数据结构的基本概念	9
1.2.1 数据结构	9
1.2.2 抽象数据类型	11
1.3 算法的基本概念	13
1.3.1 算法及算法的特性	13
1.3.2 算法的描述方法.....	14
1.4 算法分析.....	16
1.4.1 算法的时间复杂度	16
1.4.2 算法的空间复杂度	18
1.4.3 算法分析举例	18
习题 1	24

随处可见的树结构

例 4 日常生活中随处可见的树结构——行政区域



随处可见的树结构

例 5 随处可见的树结构——思维导图

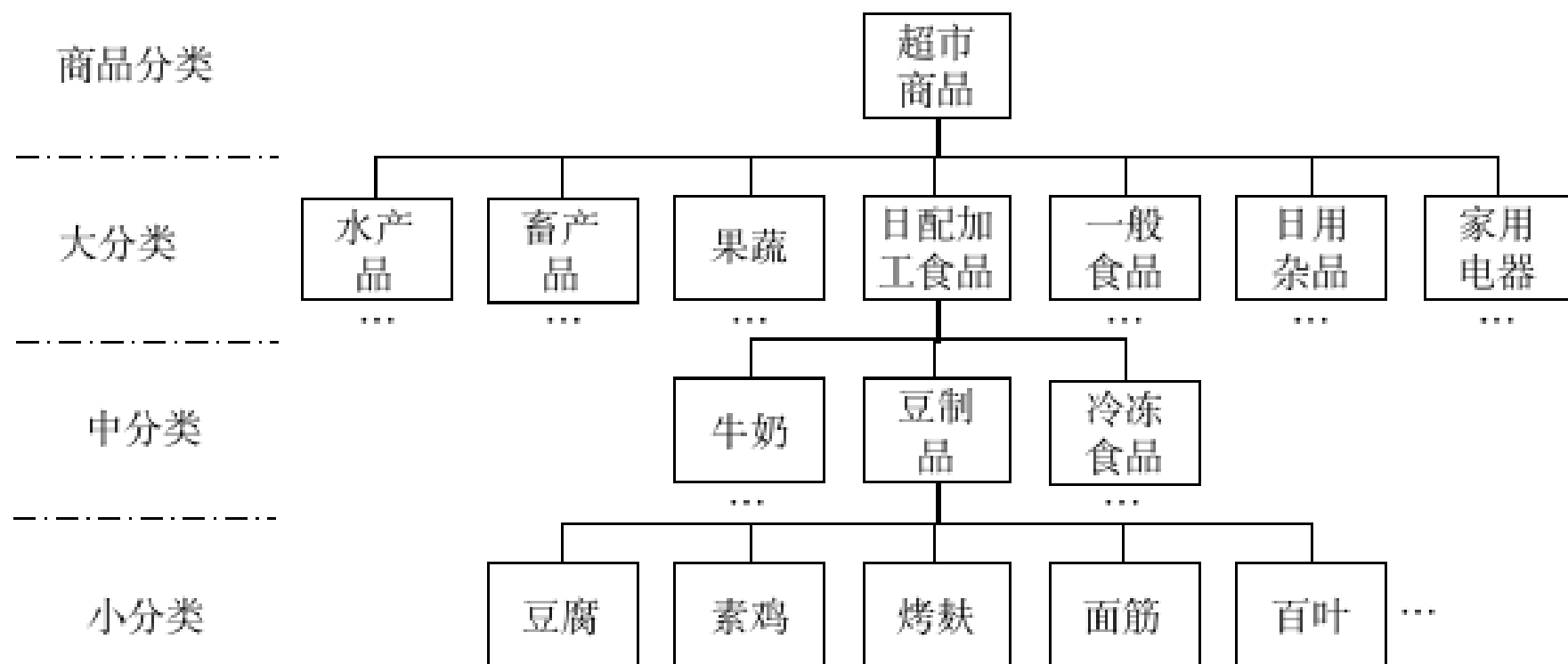


问题引入：商品分类树

问题：如何方便了解超市商品有哪些类别？每个类别可分为哪些更小的类别，以及属于该类别的终端商品有哪些？

关键：如何展示商品不同类别之间的层次关系（包含关系）

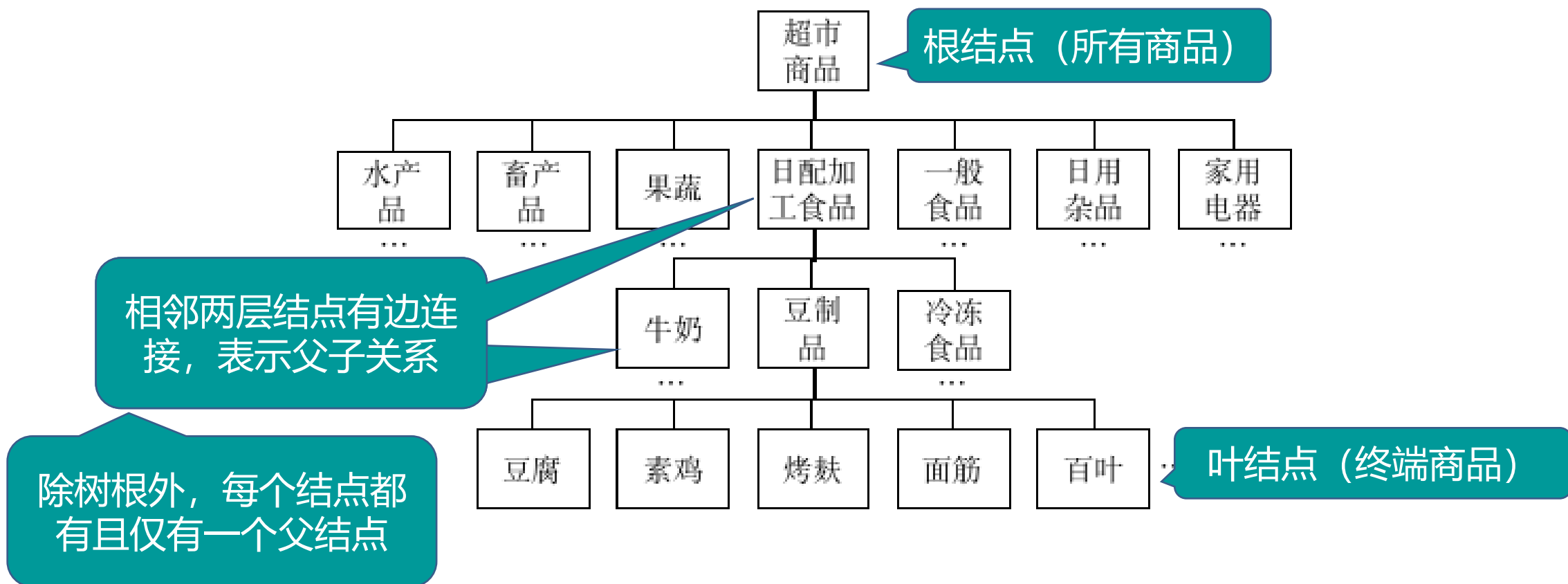
商品分类树：



问题引入：商品分类树

商品分类树

结构与倒立的树相似，展示对数以千万计的商品由上至下、由粗到细的分类



问题引入：商品分类树

商品分类树

结构与倒立的树相似，展示对数以千万计的商品由上至下、由粗到细的分类

问题

如何在计算机上表达商品分类树？能否用顺序表、链表等线性结构存储树？如何在计算机上查询每个分类包含哪些商品？如何判断两个分类之间是否有包含关系？



主要内容

树与二叉树的定义、存储实现、遍历方式及其典型应用

关于树结构



什么是树？在逻辑上有什么特点？有哪些基本术语？



如何存储树结构？



什么是二叉树？在逻辑上有什么特点？有哪些基本性质？



如何存储二叉树？



如何实现二叉树的遍历操作？



最优二叉树及应用



树



树的定义及逻辑特征



树的基本术语



树的抽象数据类型定义

树的定义与结构

树的定义:

树是由结点组成的有限集合 T , 用 $|T|$ 表示结点的数量。

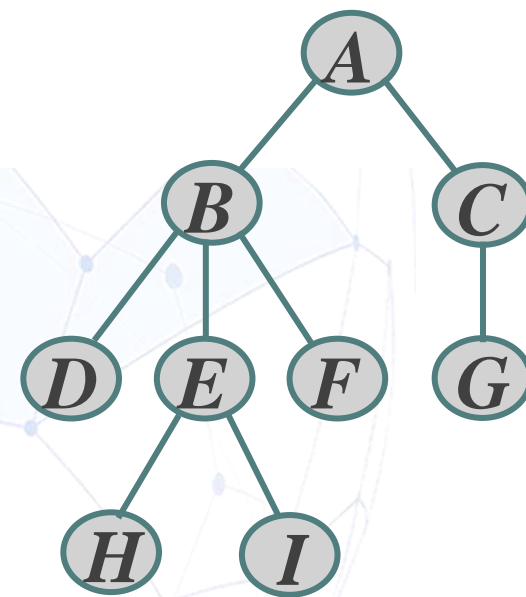
树具备以下性质:

1. $|T| = 0$, T 是空树。

2. $|T| > 0$

(1) 有且仅有一个特定的称为根的结点;

(2) 假设 T 中树根记作 $r \in T$, 其它结点 $T - \{r\}$ 划分为 $m(\geq 0)$ 个互不相交的子集 T_1, T_2, \dots, T_m , 每个子集 $T_i (i \in [1, m])$ 也是树, 称为根结点 r 的子树;



递归定义

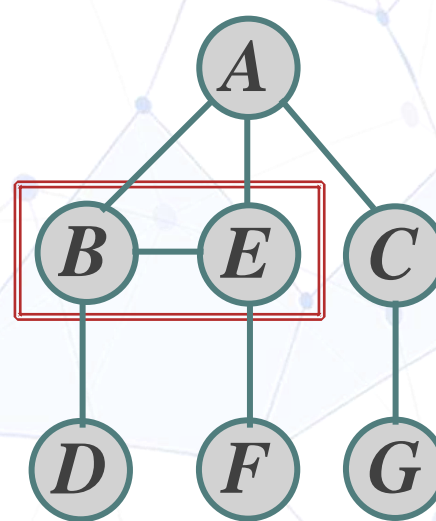
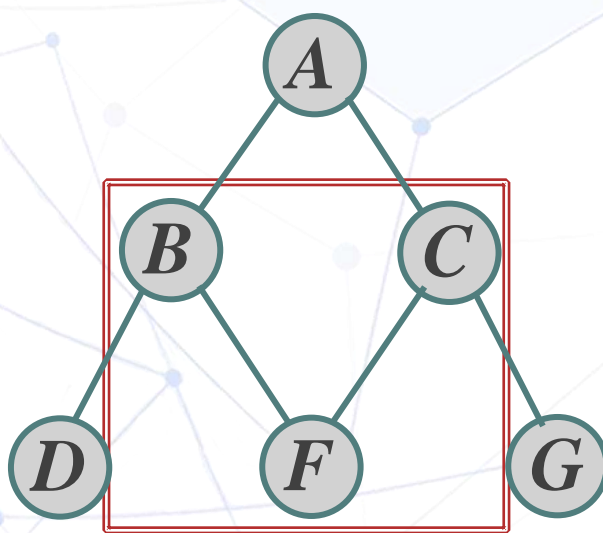
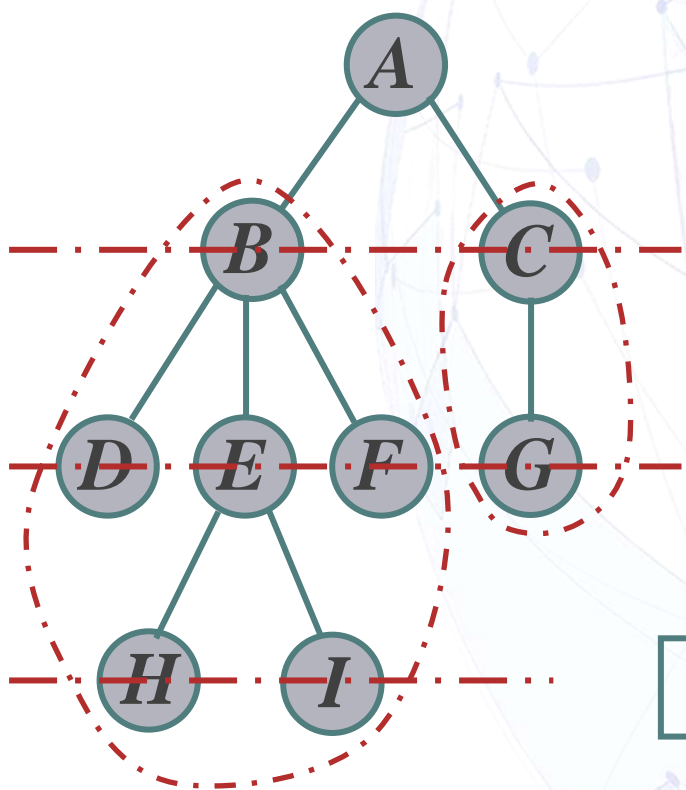
逻辑特征



互不相交的具体含义是什么？

结点：结点不能属于多个子树

边：子树之间不能有关系



互不相交



没有回路



树结构具有层次性

树的定义 (2)

(3) 如果子树 $|T_i| > 0$ ，子树的根 $r_i \in T_i$ 是 r 的子结点， r 是 r_i 的父结点。

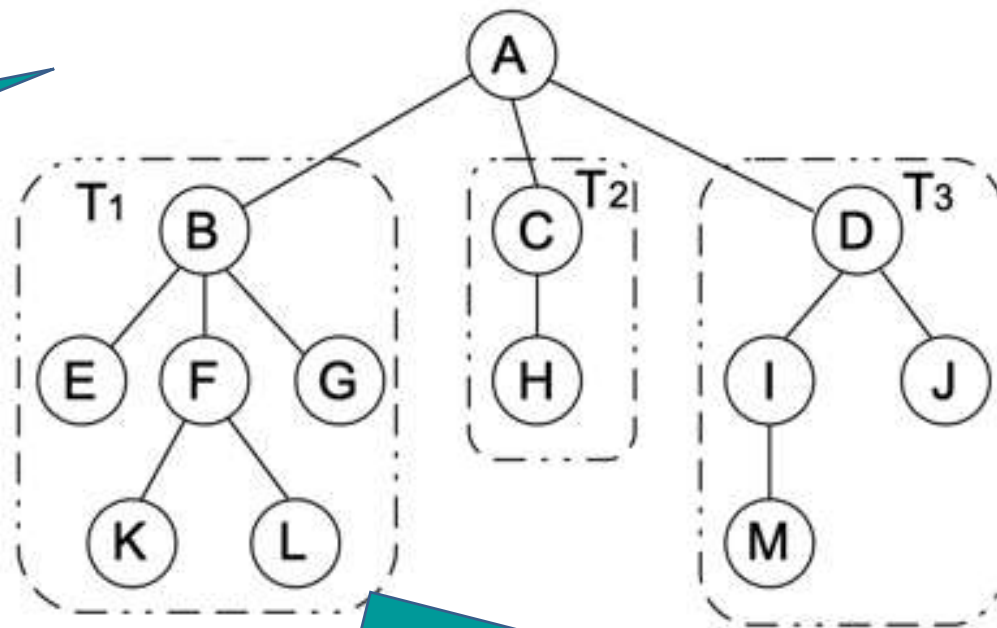
- 父子关系属于二元关系 $\langle r, r_i \rangle$
- 由于根结点没有父结点，而其它所有结点都有一个且仅一个父结点，所以由 n 个结点构成的树共包含 $n-1$ 对父子关系
- 如果用边连接有父子关系的结点，则树中共有 $n-1$ 条边

树的定义与结构

- 树的结构示例:

树 $T = \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$

- 树根: A
- 子树: $T_1 = \{B, E, F, G, K, L\}$, B为根
 $T_2 = \{C, H\}$, C为根
 $T_3 = \{D, I, J, M\}$, D为根
- 父子关系: $\langle A, B \rangle$ 、 $\langle A, C \rangle$ 、 $\langle A, D \rangle$



T_1 也是树 (递归定义)

- 树根: B
- 子树: $\{E\}$, $\{F, K, L\}$, $\{G\}$
- 父子关系: $\langle B, E \rangle$, $\langle B, F \rangle$, $\langle B, G \rangle$

树的基本术语

结点的度：子结点或非空子树的个数

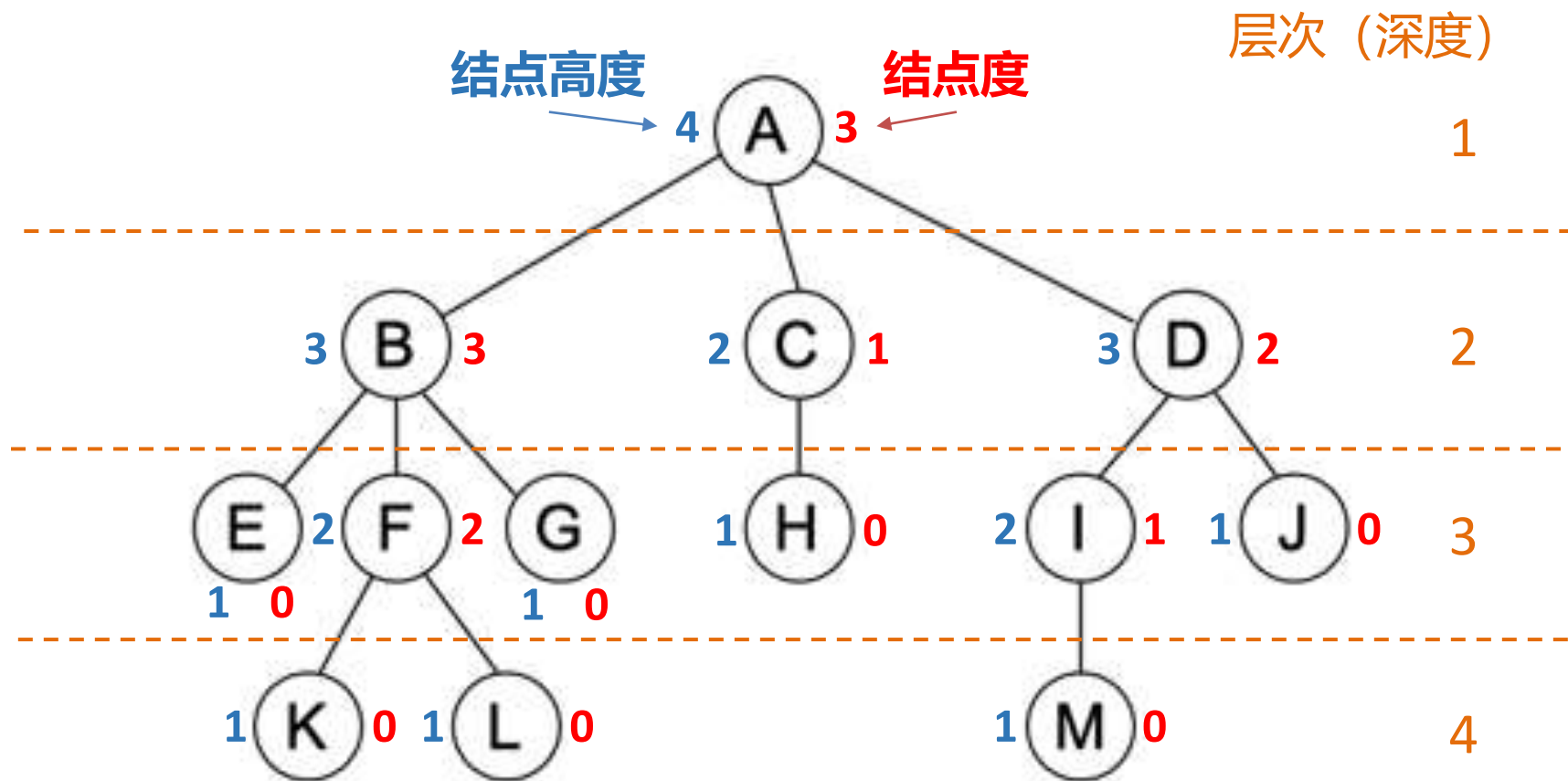
叶结点：树中度为0的结点

结点的高度：叶结点的高度等于1；中间结点的高度等于其所有子结点的高度的最大值加1

树的度：树中所有结点的度的最大值

树的高度：根结点的高度，亦称树的深度

结点的层次：根结点在第1层；如果结点的层次是 k ($k \geq 1$)，则其子结点都在第 $k+1$ 层。亦称**结点的深度**



叶结点：E, G, H, J, K, L, M

树的度：3

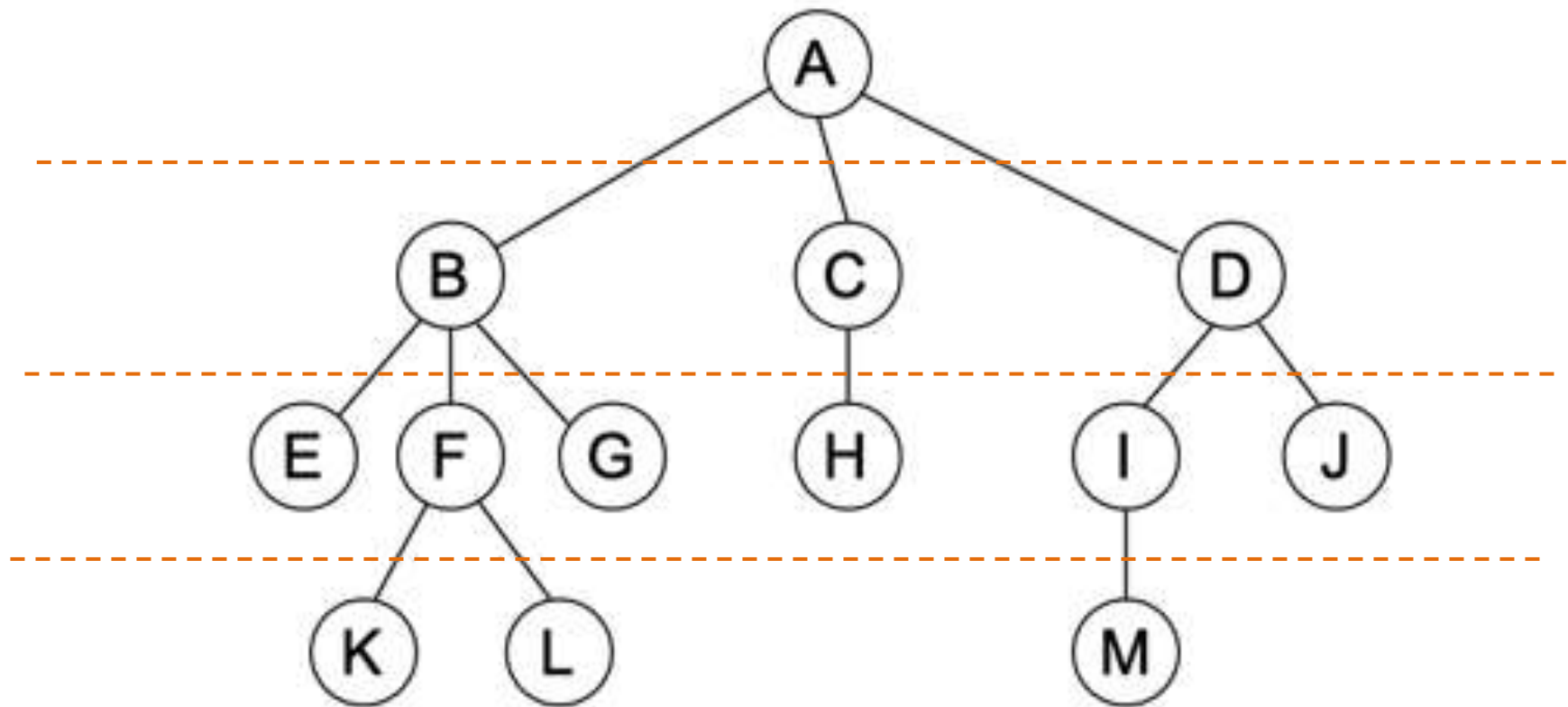
树的高度：4

树的基本术语(2)

中间结点：树中叶结点以外的结点，亦称**内部结点**，**分支节点**，**非终端结点**

兄弟结点：父结点相同的结点彼此是兄弟结点

有序树：树中各结点的子树从左向右依次排列，不能交换次序；否则称作**无序树**



中间结点：A, B, C, D, F, I

兄弟结点：{B, C, D} / {E, F, G} / {I, J} / {K, L}

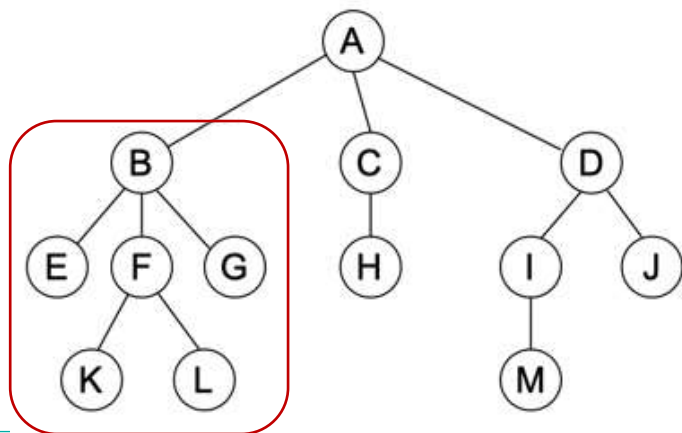


树的基本术语(3)

- **祖先结点**：根没有祖先；父结点以及父结点的祖先都是结点的祖先结点
- **子孙结点**：叶结点没有子孙；中间结点的各子结点以及子结点的子孙都是它的子孙结点

根结点是其它所有结点的祖先

以任一结点为根的子树中包含所有且仅有该结点的子孙



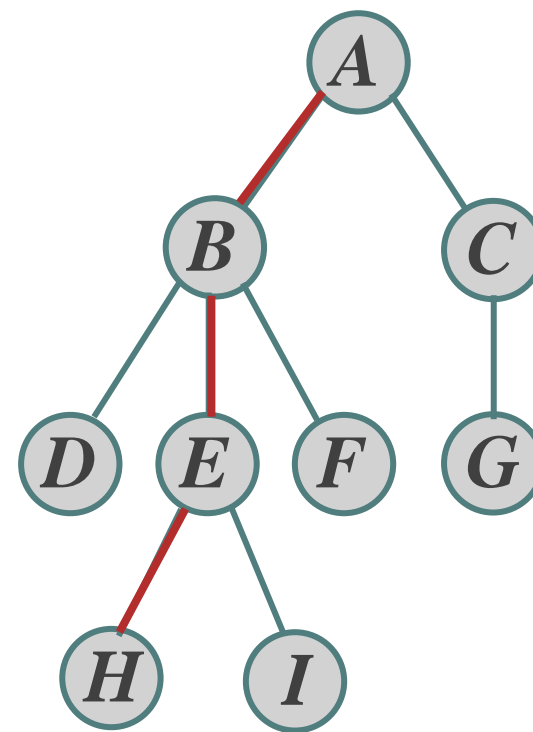
结点B的子孙：{E, F, G, K, L}

树的基本术语(4)

路径和路径长度

如果结点 u 是结点 v 的祖先，则存在路径 $\langle r_1, r_2, \dots, r_n \rangle$ ，满足 $r_1 = u$, $r_n = v$ 且 r_i 是 r_{i+1} 的父结点 ($0 < i < n$)，此时路径长度为 n

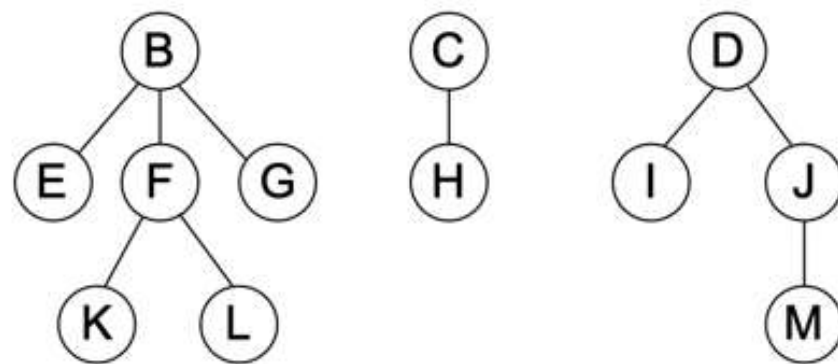
由于任何非根结点有且仅有一个父结点，且根结点只有一个，所以从祖先结点 u 到子孙结点 v 的路径是唯一的。



A是H的祖先，必然存在一条唯一路径(A,B,E,H)

树的基本术语(5)

森林： 零个或多个互不相交（独立）的树的集合



线性结构和树结构的比较



线性结构

开始结点（只有一个）：**无前驱**

终端结点（只有一个）：**无后继**

其它元素：**一个前驱,一个后继**

一对一

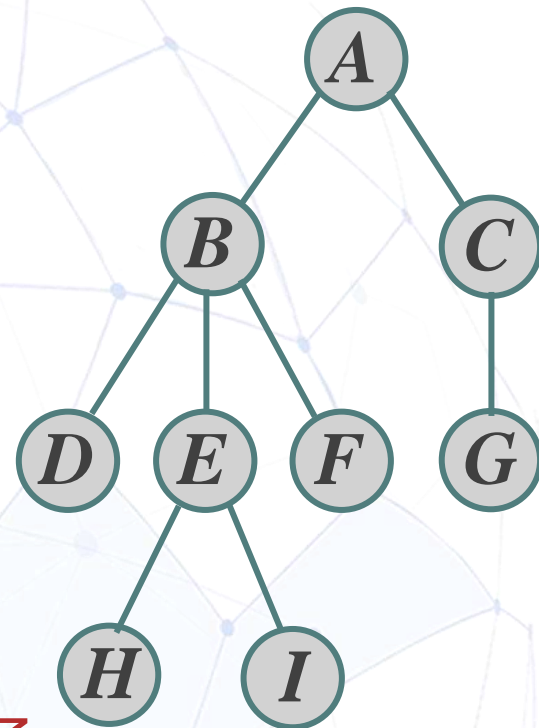
树结构

根结点（只有一个）：**无双亲**

叶子结点(可以有多个)：**无孩子**

其它结点：**一个双亲,多个孩子**

一对多



树的抽象数据类型定义

ADT Tree

DataModel

树由一个根结点和若干棵子树构成，树中结点具有层次关系

Operation

树的基本操作

endADT

树的应用很广泛，在不同的实际应用中，树的基本操作不尽相同

树的基本操作

- $\text{InitTree}(tree)$: 初始化一个空树 $tree$
- $\text{CreatTree}(tree, definition)$: 按照 $definition$ 构造一个树
- $\text{IsEmpty}(tree)$: 树 $tree$ 为空返回**true**, 否则返回**false**
- $\text{Root}(tree)$: 返回树 $tree$ 的根结点
- $\text{Get}(tree, node)$: 返回树 $tree$ 的结点 $node$ 的值
- $\text{Parent}(tree, node)$: 返回树 $tree$ 中结点 $node$ 的父结点
- $\text{GetChild}(tree, node, k)$: 返回树 $tree$ 中结点 $node$ 的第 k 个子树
- $\text{InsertChild}(tree, node, k, subtree)$: 将树 $subtree$ 插入到树 $tree$ 中, 使其成为结点 $node$ 的第 k 个子树
- $\text{Search}(tree, x)$: 在树 $tree$ 中查找值为 x 的结点, 如果查找成功, 返回结点, 否则返回NIL
- $\text{Traverse}(tree)$: 访问树 $tree$ 中每个结点, 且每个结点只访问一次

在讲完二叉树后再讲树的基本操作

1. 在树结构中，逻辑关系体现为兄弟之间的关系。

☐ A 正确

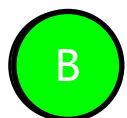
☒ B 错误

提交

2. 在树结构中，根结点只有一个，但是一定有多个叶子结点。



正确



错误

提交