

第5章 树和二叉树

5.1 本章教学要点

5.1.1 知识点的组织结构

树结构是非常重要的非线性结构，具有严格的层次特征，是本课程的重点和难点。本章的内容由树和二叉树两部分组成，并且以二叉链表存储结构为媒介，实现了树和二叉树之间的转换，知识点的组织结构如图 5-1 所示。

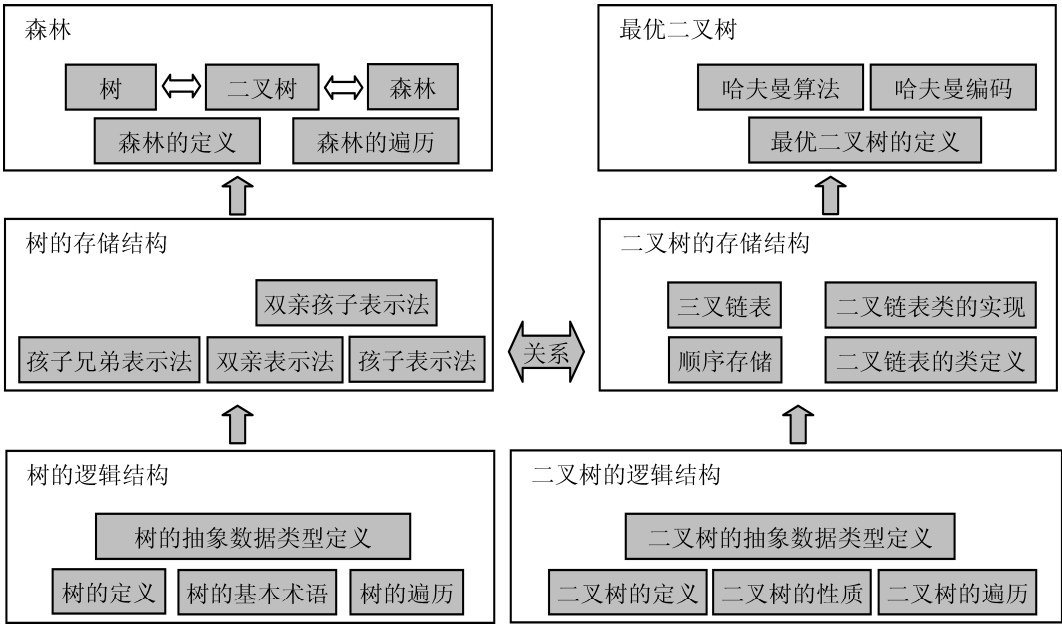


图 5-1 本章知识点的组织结构

5.1.2 教学提示

本章的教学重点是：①二叉树的性质；②二叉树和树的存储表示；③二叉树的遍历及算法实现；④树与二叉树的转换关系；⑤哈夫曼树及应用。

本章的教学难点是：①二叉树遍历算法的非递归实现；②基于二叉树的遍历实现二叉树的其他操作；③树的基本操作的实现。

对于树的教学要抓住一条明线：树的逻辑结构→树的存储结构，一个重点：树的遍历操作。对于树的逻辑结构，要从树的定义出发，在与线性表定义进行比较的基础上，把握要点理解树的定义及其逻辑

特征,通过具体实例理解树的基本术语,从逻辑上理解树的遍历操作,最后给出树的抽象数据类型定义。对于树的存储结构,要以如何表示树中结点之间的逻辑关系为出发点,掌握树的不同存储方法以及它们之间的关系。

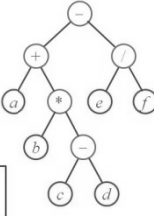
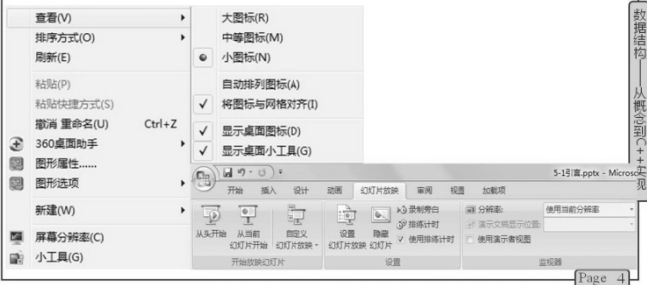
对于二叉树的教学要抓住一条明线:二叉树的逻辑结构→二叉树的存储结构→二叉树的实现,一个重点:二叉树的遍历操作及其实现。对于二叉树的逻辑结构,要从二叉树的定义出发,在与树的定义进行比较的基础上,理解树和二叉树是两种树结构,通过二叉树的性质加深对二叉树逻辑结构的理解,从逻辑上掌握二叉树的遍历方法,最后给出二叉树的抽象数据类型定义。对于二叉树的存储结构,要从二叉树的逻辑特征和基本性质出发,掌握二叉树的不同存储方法以及它们之间的关系,基于二叉链表存储结构讨论二叉树遍历操作的实现。

5.2 教学专题 1——引言

5.2.1 教学过程设计

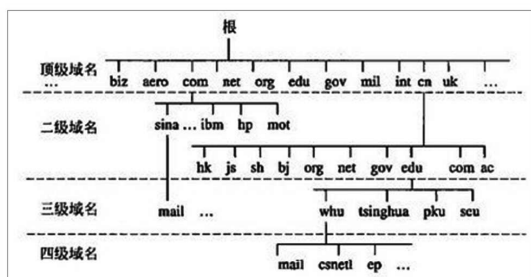
教学专题	树的提出	授课学时	15 分钟			
教学章节	5.1	授课对象	计算机专业本科生			
教学课型	理论课	授课形式	课堂讲授			
教学重点	树结构作为数据模型的应用实例					
教学难点	无					
教学内容和 教学目标	知识点		学习要求			
			了解	理解	掌握	熟练掌握
	实际问题中树结构的例子			√		
	计算机系统中树结构的例子			√		
	生活中树结构的例子		√			
教学过程	教学过程设计					时间分配
	1. 以操作系统的文件目录结构为例，说明树在计算机系统的使用。					3 分钟
	2. 介绍二叉表示树，给出一个表达式转换为二叉表示树的例子。					6 分钟
	3. 介绍计算机系统中树结构的经典实例。					3 分钟
	4. 介绍日常生活中树结构的经典实例					3 分钟
教学提示	很多实际问题抽象的数据模型是树结构，最简单的树结构是二叉树，说明有必要忽略具体的数据类型，讨论如何存储和实现树和二叉树					
媒体使用	多媒体课件：借助视觉通道，用图片及动画形象展示实际问题中、计算机系统和日常生活中随处可见的树结构					
教学后记	教学感想、意外发现、点滴收获、个别疏漏及补救					

5.2.2 课件及教学设计

 第五章 树和二叉树 5-1 树的提出	 电子课件
文件系统 例 1 操作系统的文件目录结构  许多问题抽象出的数据模型具有层次关系 Page 2	树结构通常用来描述具有包含关系或层次关系的数据模型，在计算机系统和日常生活中都有很多应用实例。树与二叉树是本课程的难点和重点。 操作系统用树结构来组织文件夹和文件，因此，不同文件夹下可以有同名文件。
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  Page 3	1929 年波兰数学家提出表达式的一种前缀表示法，可以不用考虑优先级直接计算，例如，$3+4\times 2$ 的前缀表达式是 $+3\times 42$，从前向后扫描遇到运算符就与随后的两个操作数进行计算。为了便于计算机处理，避免重复扫描表达式，提出了后缀表达式，也称逆波兰式。
随处可见的树结构 例 3 计算机系统上随处可见的树结构——菜单  Page 4	每个菜单与其下拉菜单是树结构，右键的弹出菜单也是树结构。你认为菜单树结构的层数最多是多少？

随处可见的树结构

例4 计算机系统中随处可见的树结构——internet的名字空间

数据结构
从概念到C++实现

Page 5

在因特网名字空间对应的树结构中，一条路径构成一个实际网址，例如，tsinghua.edu.cn。

随处可见的树结构

例5 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

数据结构
从概念到C++实现

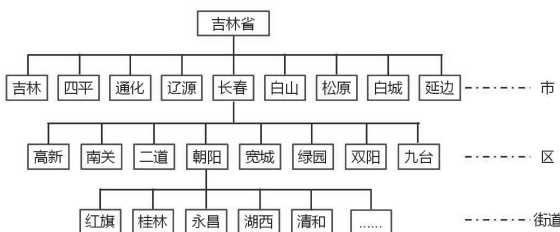
Page 6

对于一本书或一篇论文，什么是结构合理？首先看目录，不存在度为1的分支结点，章和章之间、节和节之间、小节和小节之间是并列或递进关系，章、节和小节之间是包含关系。

读小说时，复杂的家族关系用树结构来描述就清楚多了。

随处可见的树结构

例6 日常生活中随处可见的树结构——行政区域

数据结构
从概念到C++实现

Page 7

树结构在行政管理中有很多应用，例如，一个单位的组织机构、行政区域的划分等。

随处可见的树结构

例7 随处可见的树结构——知识树

数据结构
从概念到C++实现

Page 8

中小學生都用知識樹來梳理知識單元，使零散的知識用樹結構串接起來。

随处可见的树结构

例 8 随处可见的树结构——思维导图

Page 9

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

Page 10

近年来流行的思维导图也是树结构，能够很好地梳理思路 and 要表达的事情，例如，找工作时投递的简历，用思维导图写到一页纸上，既方便 HR 阅读也便于迅速了解求职者。

树结构较复杂，需要介绍一些常见的的基本术语。

二叉树是最简单的树结构，且树可以转换为二叉树，因此重点讨论二叉树，包括逻辑结构、基本性质、存储结构、基本操作的实现。

5.3 教学专题 2——树的逻辑结构

5.3.1 教学过程设计

教学专题	树的逻辑结构	授课学时	45 分钟			
教学章节	5.2	授课对象	计算机专业本科生			
教学课型	理论课	授课形式	课堂讲授			
教学重点	树的定义和基本术语；树的遍历操作					
教学难点	树的遍历操作					
教学内容和 教学目标	知识点	学习要求				
		了解	理解	掌握	熟练掌握	
	树的定义				√	
	树的基本术语			√		
	树的抽象数据类型定义		√			
	树的遍历操作				√	

	教学过程设计	时间分配
教学过程	1. 给出树的定义，在与线性表定义比较的基础上，抓住要点——互不相交，深刻理解树的逻辑特征。	5 分钟
	2. 结合实例分类讲授树的基本术语。	20 分钟
	3. 小结：将树结构和线性结构从逻辑结构上做比较。	3 分钟
	4. 复习 ADT 的三个视图，给出树的 ADT 定义。	5 分钟
	5. 理解什么是遍历，重点解释两点：访问的含义和遍历次序。	3 分钟
	6. 由树的组成得到两种遍历次序：前序遍历和后序遍历，由树的层次特征得到另一种遍历次序：层序遍历。	2 分钟
	7. 树的遍历操作定义采用递归方法。引申：树结构本身具有递归特性，相应地，求解问题常常采用递归的方法。	4 分钟
	8. 结合树的实例练习各种遍历方法	3 分钟
教学提示	树结构在计算机软件系统中的应用较多，比较直观的有操作系统中的文件组织、软件界面中的多级下拉菜单，另外，比较典型的应用还有描述递归执行过程的递归树、描述判定过程的判定树、编译程序中进行语法检查的语法树等。注意从应用的角度以具体实例引起学生对本章的重视。 对于树的遍历，首先要从逻辑上理解遍历操作的执行过程，然后再基于存储结构实现，这也体现了抽象分级的观点	
媒体使用	多媒体课件：用动画辅助讲授基本术语，形象展示树的遍历过程。 扫描二维码获得多媒体课件和程序源代码	
课后导读	树结构更复杂的抽象数据类型定义，请参见《数据结构（C 语言版）》（严蔚敏编著，清华大学出版社）P119；除了逻辑关系图，树结构还有其他逻辑表示方法，请参见《数据结构（C 语言版）》（严蔚敏编著，清华大学出版社）P120	
教学后记	教学感想、意外发现、点滴收获、个别疏漏及补救	

5.3.2 课件及教学设计





电子课件

讲什么？

- 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
- 树的遍历的操作定义

数据结构——从概念到 C++ 实现

Page 2

树的基本术语要达到识记（认识并记住）的程度，注意理解而不是死记硬背。

本讲在不涉及存储结构的前提下，对于树的遍历讨论其操作定义并运行实例，注意是在逻辑结构上讨论树的遍历操作，也就是“问题→想法”阶段。

树的定义

数据元素



- ★ 树： n ($n \geq 0$) 个结点的有限集合，当 $n=0$ 时，称为空树；任意一棵非空树 T 满足以下条件：
- (1) 有且仅有一个特定的称为根的结点；
 - (2) 当 $n > 1$ 时，除根结点之外的其余结点被分成 m ($m > 0$) 个互不相交的有限集合 $T_1, T_2, \dots, T_m$ ，其中每个集合又是一棵树，并称为这个根结点的子树。

树的定义是采用递归方法

数据结构——从概念到 C++ 实现

Page 3

与线性表的定义比较，最大的差别是什么？序列——集合，逻辑结构不同。

树的定义表明树结构本身具有递归特征，相应地，求解问题常常采用递归的方法。递归技术是一些同学尚未很好掌握的一种程序设计技术，本课程将进一步理解和掌握递归技术。

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

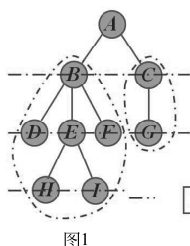


图1

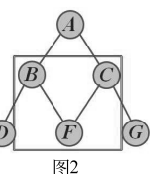


图2

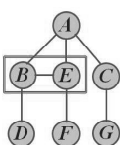


图3

互不相交 $\Rightarrow$ 没有回路 $\Rightarrow$ 树结构具有层次性

数据结构——从概念到 C++ 实现

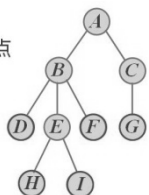
Page 4

如何理解子集的互不相交？（1）在图 2 中，结点 F 属于哪棵子树？（2）在图 3 中，结点 B 和 E 是什么关系？

略微观察就会发现，由于子树互不相交，树中必然没有回路，因此树结构具有层次性。

基本术语

- ★ 结点的度：结点所拥有的子树的个数
- ★ 树的度：树中各结点度的最大值
- ★ 叶子结点：度为 0 的结点，也称为终端结点
- ★ 分支结点：度不为 0 的结点，也称为非终端结点



数据结构——从概念到 C++ 实现

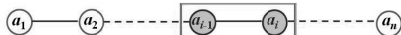
Page 5

理解起来，结点的度就是从该结点出发的分枝数（即边数）。

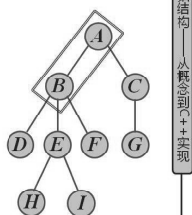
叶子结点有哪些？分支结点有哪些？根结点是分支结点吗？

基本术语

- ★ 孩子：树结构中某结点的子树的根结点称为这个结点的孩子结点
- ★ 双亲：这个结点称为它孩子结点的双亲结点
- ★ 兄弟：具有同一个双亲的孩子结点互称为兄弟



- 📎 在线性结构中，逻辑关系表现为前驱——后继
- 📎 在树结构中，逻辑关系表现为双亲——孩子



数据结构——从概念到实现

Page 6

强调：双亲和孩子体现了树结构中结点之间的逻辑关系。

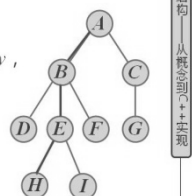
观察线性表和树的逻辑结构图，逻辑关系体现为边，其他关系如兄弟都是间接关系。

从术语可以发现，家谱是一种典型的树结构，结合家谱理解树的基本术语。

基本术语

- ★ 路径：结点序列 $n_1, n_2, \dots, n_k$ 称为一条由 n_1 至 n_k 的路径，当且仅当满足如下关系：结点 n_i 是 n_{i+1} 的双亲 ($1 \leq i < k$)
- ★ 路径长度：路径上经过的边的个数
- ★ 祖先、子孙：如果有一条路径从结点 x 到结点 y ，则 x 称为 y 的祖先，而 y 称为 x 的子孙

- 📎 在树结构中，路径是唯一的



数据结构——从概念到实现

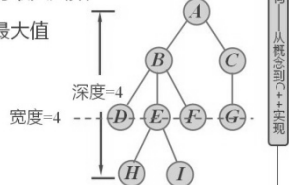
Page 7

由于限定结点 n_i 是 n_{i+1} 的双亲，因此路径都是自上而下的，例如，通常不认为 HEB 是一条路径。强调路径用结点序列来表达。

显然，某结点的子树中所有结点都是该结点的子孙。

基本术语

- ★ 结点所在层数：根结点的层数为 1；对其余结点，若某结点在第 k 层，则其孩子结点在第 $k+1$ 层
- ★ 树的深度（高度）：树中所有结点的最大层数
- ★ 树的宽度：树中每一层结点个数的最大值



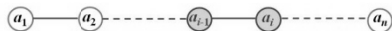
数据结构——从概念到实现

Page 8

为什么树可以定义层数？子树互不相交。

有些教材对于树的深度和高度的定义不同；有些教材将根结点所在的层数定义为 0。

线性结构和树结构的比较



线性结构

树结构

- | | |
|----------------|-----------------|
| 开始结点（只有一个）：无前驱 | 根结点（只有一个）：无双亲 |
| 终端结点（只有一个）：无后继 | 叶子结点（可以有多个）：无孩子 |
| 其它元素：一个前驱，一个后继 | 其它结点：一个双亲，多个孩子 |
| 一对一 | 一对多 |

数据结构——从概念到实现

Page 9

将树结构和线性结构在逻辑结构上进行比较，通过对比将知识点串接起来。

抽象数据类型定义

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

```

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
    InitTree: 初始化一棵树
    DestroyTree: 销毁一棵树
    PreOrder: 前序遍历树
    PostOrder: 后序遍历树
    LeverOrder: 层序遍历树
endADT
  
```

简单起见，只讨论树的遍历

Page 10

如何定义树的 ADT 呢？为了突出主题，不失一般性，只讨论遍历操作，更简单的操作（如查找结点 x 的双亲、为结点 x 增加一个孩子 y 等）在讲授存储结构时通过课堂练习完成，更复杂的操作（如统计结点个数）在遍历过程中完成。

树的遍历



什么是遍历？线性结构如何遍历？

简言之，遍历是对数据集进行没有遗漏、没有重复的访问



★ 树的遍历：从根结点出发，按照某种次序访问树中所有结点，并且每个结点仅被访问一次

前序（根）、后序（根）和层序（次）等

抽象操作，可以是对结点进行的各种处理，这里简化为输出结点的数据

Page 11

由于元素之间具有唯一的前驱和后继关系，线性表可以按序号依次访问每个数据元素。

如何理解访问？访问是一种抽象操作，即逻辑操作、不确定的操作，实际应用中可以是对结点进行的各种处理。体会抽象的好处。

根结点相当于开始结点；按照什么次序呢？考虑树的组成。

前序遍历



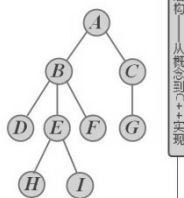
★ 树的前序遍历操作定义：

若树为空，则空操作返回；否则

(1) 访问根结点

(2) 从左到右前序遍历根结点的每一棵子树

前序遍历序列：A BDEHIF CG
 $T_1 \quad T_2$



Page 12

由树的定义出发，分析树的构成 $(R, T_1, \dots, T_m)$ ，采取问题分解的方法——分治法求解问题。以子树而不是结点为单位进行遍历，引导学生的框架（系统）思维。

树的前序遍历是递归的，根本原因是什么？树结构本身具有递归特征。

后序遍历



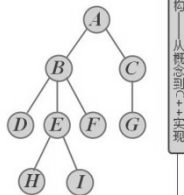
★ 树的后序遍历操作定义：

若树为空，则空操作返回；否则

(1) 从左到右后序遍历根结点的每一棵子树

(2) 访问根结点

后序遍历序列：DHIEFB GC A
 $T_1 \quad T_2$



Page 13

可以看出是直接递归，注意访问结点的框架：先访问第 1 棵子树的 6 个结点，再访问第 2 棵子树的 2 个结点，最后访问根结点，不要把遍历序列讲零散了。

