


 普通高等教育计算机规划教材

数据结构与算法 (Java版)

罗文劼 王 苗 张小莉 编著

 提供电子教案和习题解答



下载网址 <http://www.cmpedu.com>

 机械工业出版社
CHINA MACHINE PRESS



013063217

TP311.12-43
176

普通高等教育计算机规划教材

数据结构与算法 (Java 版)

罗文劫 王 苗 张小莉 编著



TP311.12-43
176



机械工业出版社



北航 C1671325

013083517

本书共包括8章内容,详细讲述了线性结构、树结构和图结构中的数
据表示及数据处理的方法,并对查找和排序两种重要的数据处理技术作了
细致的探讨。书中对每一类数据结构的分析均按照“逻辑结构—存储结
构—基本运算的实现—时空性分析—典型例题—知识点小结—扩展学习导
读—练习题—实验题”的顺序来进行,算法全部采用Java语言描述,很
容易转换成程序。本书语言叙述通俗易懂,由浅入深,算法可读性好,应
用性强。书中还配有大量算法设计的例子,便于读者理解和掌握数据结
构中数据表示和数据处理的方法。

本书可作为高等院校计算机和信息类相关专业“数据结构”课程的教
材,也可作为高职高专同类专业的教学用书及各类工程技术人员的参考用书。

本书配有电子教案和习题解答,需要的教师可登录 www.cmpedu.com
免费注册、审核通过后下载,或联系编辑索取(QQ: 2399929378, 电
话: 010-88379753)。

图书在版编目(CIP)数据

数据结构与算法: Java版/罗文劼,王苗,张小莉编著. —北京:机械工
业出版社,2013.8

普通高等教育计算机规划教材

ISBN 978-7-111-42690-5

I. ①数… II. ①罗… ②王… ③张… III. ①数据结构-高等学校
-教材 ②算法分析-高等学校-教材 ③JAVA语言-程序设计-高等
学校-教材 IV. ①TP311.12 ②TP312

中国版本图书馆CIP数据核字(2013)第115443号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

责任编辑:和庆娣

责任印制:张楠

涿州市京南印刷厂印刷

2013年7月第1版·第1次印刷

184mm×260mm·20印张·496千字

0001-3000册

标准书号:ISBN 978-7-111-42690-5

定价:43.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

教材网:<http://www.cmpedu.com>

销售一部:(010)68326294

机工官网:<http://www.cmpbook.com>

销售二部:(010)88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010)88379203

封面无防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基础素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

- 1) 反映计算机技术领域的新发展和新应用。
- 2) 为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、多媒体光盘、课程设计和毕业设计指导等内容。
- 3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。
- 4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- 5) 注重教材的实用性、通用性，适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

希望计算机教育界的专家和老师们能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

“数据结构”是计算机及相关专业的一门重要的专业基础课，是介于“数学”、“计算机硬件”和“计算机软件”之间的一门计算机科学与技术领域的核心课程，同时数据结构技术也被广泛应用于信息科学、系统工程、应用数学以及各种工程技术领域。该课程主要介绍如何合理地组织和表示数据、如何有效地存储和处理数据、如何正确地设计算法以及对算法的优劣进行分析和评价。

在数据结构的教材中，对算法的描述采用 C 语言和 C++ 的较多，而采用 Java 语言描述的较少。随着软件开发技术的发展，Java 语言作为完全面向对象的语言，已成为当前应用开发中使用最广泛的语言之一。因此，采用 Java 语言描述数据结构会为 Java 语言编程人员提供更实用的参考。

为了适应一些高校对数据结构 Java 版的需求，我们在机械工业出版社《数据结构与算法》（第 2 版）C 语言版的基础上编写了本书。本书以“面向应用，易教易学”为指导，并在以下几方面有所改进。

1) 章节结构的调整。将线性表、栈和队列、串、数组和广义表等与线性结构相关的内容编写在线性结构一章中，本书按照绪论、算法设计用到的递归技术、线性结构、树结构、图结构、查找技术、排序技术以及扩展应用划分章节，组织教材内容，内容规整，简洁明了。

2) 应用性强的内容。将基础性、实用性的软件开发技术写入教材，略去一些理论推导和烦琐的数学证明，同时也删掉了平时讲不到、难度较大或应用性差的一些问题，增加了部分更基础、更常用的或应用性强的内容。

3) 问题引入的方式。主要章节的开始采用问题驱动引入，从常识性或典型问题入手，引导读者思考，使读者更快、更自然地进入到内容的学习中。

4) 有针对性的示例。在每一章讲解基本知识之后，都列举一些对应的应用问题，给出典型例题的分析与解决，帮助读者理解和掌握本章节的知识点在实践中的运用方法。

5) 丰富的配套练习。每章除了理论课教学内容外，还包括练习题、实验题，帮助学生全面掌握所要求的知识点。本书最后还给出实验要求、模拟试卷、部分参考书目和参考网站，为读者提供实验课程的指导和辅助学习的资料。另外，在每一章的结尾有对本章知识点的总结和扩展学习的阐述，既能帮助读者回顾本章的内容，掌握学习重点，又能为有需要进一步提高的读者提供相关的学习索引。

本书由河北大学的罗文劼教授组织并统稿，张小莉教授审稿。其中第 1、3 章由张小莉和罗文劼共同编写，第 4、5、8 章由罗文劼编写，第 2、6、7 章由王苗编写。

在本书的编写过程中，刘宇对 Java 语言描述算法的规范性给出了有益的建议，石强、苗秀芬、王硕等对此书的编写提出了有益的意见和建议，在此一并表示感谢。

由于编者水平有限，书中难免存在疏漏之处，恳请读者批评指正。

编 者

目 录

出版说明

前言

第 1 章 数据结构与算法 1

1.1 引言 1

1.1.1 为什么要学习数据结构 1

1.1.2 数据结构课程的内容 4

1.2 数据结构的概念 4

1.2.1 基本概念和术语 4

1.2.2 抽象数据类型 7

1.3 数据结构的描述方法 8

1.3.1 数据结构的 C 语言描述 8

1.3.2 数据结构的 C++ 语言描述 9

1.3.3 数据结构的 Java 语言描述 10

1.4 算法 11

1.4.1 算法及其特征 11

1.4.2 算法的描述 12

1.4.3 算法的性能分析 12

1.5 典型例题 15

1.6 本章小结 16

1.6.1 知识点小结 16

1.6.2 扩展学习导读 17

练习题 17

实验题 18

题目 1 比较算法复杂性描述函数的

增长 18

题目 2 矩阵连乘算法的时间和空间

复杂度 18

第 2 章 递归 19

2.1 引言 19

2.1.1 问题的提出 19

2.1.2 什么是递归 19

2.2 递归调用的实现原理 20

2.2.1 递归算法的构成 20

2.2.2 递归调用的内部过程 21

2.2.3 递归的正确性证明 22

2.3 递归转化为非递归 22

2.3.1 递归转化为递推 22

2.3.2 递归转化为回溯 23

2.3.3 手工模拟递归 23

2.4 典型例题 24

2.5 本章小结 26

2.5.1 知识点小结 26

2.5.2 扩展学习导读 26

练习题 27

实验题 27

题目 1 全排列的递归实现 27

题目 2 八皇后问题 28

题目 3 比较递归与非递归算法的

时空效率 28

第 3 章 线性结构 29

3.1 引言 29

3.1.1 问题的提出 29

3.1.2 线性表的定义 30

3.1.3 线性表的基本运算 30

3.2 线性表的顺序存储与实现 31

3.2.1 顺序表 31

3.2.2 顺序表上基本运算的实现 33

3.2.3 顺序表应用举例 36

3.3 线性表的链式存储与实现 38

3.3.1 单链表 38

3.3.2 单链表上基本运算的实现 40

3.3.3 循环链表 46

3.3.4 双向链表 47

3.3.5 链表应用举例 49

3.4 顺序表和链表的比较 52

3.5 堆栈 53

3.5.1 堆栈的定义及基本运算	53	4.4.3 统计给定二叉树中叶子结点的数目	127
3.5.2 堆栈的存储及运算实现	53	4.4.4 表达式运算	128
3.5.3 堆栈的应用举例	57	4.4.5 由遍历序列恢复二叉树	128
3.6 队列	66	4.5 线索二叉树	130
3.6.1 队列的定义及基本运算	66	4.5.1 线索二叉树的定义及其结构	130
3.6.2 队列的存储及运算实现	67	4.5.2 线索二叉树的创建	132
3.6.3 队列的应用举例	73	4.5.3 线索二叉树的遍历	133
3.7 其他线性结构及扩展	76	4.6 最优二叉树	136
3.7.1 字符串	76	4.6.1 最优二叉树的概念	136
3.7.2 数组	81	4.6.2 最优二叉树的构造	138
3.7.3 特殊矩阵	83	4.6.3 最优二叉树的应用——哈夫曼编码	140
3.7.4 稀疏矩阵	86	4.7 树	143
3.8 典型例题	90	4.7.1 树的基本操作	143
3.9 本章小结	100	4.7.2 树的表示	143
3.9.1 知识点小结	100	4.7.3 树的存储	144
3.9.2 扩展学习导读	102	4.7.4 树和森林与二叉树之间的转换	148
练习题	103	4.7.5 树或森林的遍历	150
实验题	105	4.7.6 树的应用	151
题目1 Josephus 环问题	105	4.8 典型例题	152
题目2 一元多项式运算	106	4.9 本章小结	163
题目3 模拟停车场管理	107	4.9.1 知识点小结	163
第4章 树结构	109	4.9.2 扩展学习导读	165
4.1 引言	109	练习题	166
4.1.1 问题提出	109	实验题	168
4.1.2 相关概念	110	题目 哈夫曼编码/译码器	168
4.2 二叉树	113	第5章 图结构	170
4.2.1 二叉树的基本运算	113	5.1 引言	170
4.2.2 二叉树的主要性质	114	5.1.1 问题的提出	170
4.2.3 二叉树的存储	115	5.1.2 相关概念	171
4.2.4 二叉树基本运算的实现	118	5.1.3 图的基本操作	173
4.3 二叉树的遍历	120	5.2 图的存储	174
4.3.1 递归方法实现二叉树遍历	120	5.2.1 邻接矩阵	174
4.3.2 非递归方法实现二叉树的三种遍历	122	5.2.2 邻接表	176
4.3.3 按层次遍历二叉树	125	5.3 图的遍历	179
4.4 二叉树遍历的应用	126	5.3.1 深度优先搜索	179
4.4.1 构造二叉树的二叉链表存储	126	5.3.2 广度优先搜索	181
4.4.2 在二叉树中查找值为 x 的数据元素	127		

5.3.3 遍历图的简单应用	183	6.7 本章小结	251
5.4 生成树和最小生成树	184	6.7.1 知识点小结	251
5.4.1 生成树和生成森林	184	6.7.2 扩展学习导读	252
5.4.2 最小生成树	185	练习题	253
5.4.3 构造最小生成树的 Prim 算法	186	实验题	254
5.4.4 构造最小生成树的 Kruskal 算法	189	题目1 职工信息检索系统	254
5.5 最短路径	191	题目2 个人图书管理系统	255
5.5.1 单源点最短路径——Dijkstra 算法	192	第7章 排序技术	256
5.5.2 每一对顶点之间的最短路径	195	7.1 引言	256
5.6 拓扑排序与关键路径	196	7.1.1 问题的提出	256
5.6.1 有向无环图的概念	196	7.1.2 相关概念	256
5.6.2 AOV 网与拓扑排序	197	7.2 插入排序	258
5.6.3 AOE 网与关键路径	200	7.2.1 直接插入排序	258
5.7 典型例题	205	7.2.2 折半插入排序	259
5.8 本章小结	215	7.2.3 希尔排序	260
5.8.1 知识点小结	215	7.3 交换排序	261
5.8.2 扩展学习导读	217	7.3.1 冒泡排序	262
练习题	218	7.3.2 快速排序	263
实验题	219	7.4 选择排序	265
题目 校园导游程序	219	7.4.1 简单选择排序	265
第6章 查找技术	221	7.4.2 树结构选择排序	266
6.1 引言	221	7.4.3 堆排序	267
6.1.1 问题提出	221	7.5 归并排序	270
6.1.2 相关概念	221	7.6 基数排序	273
6.2 线性表查找	222	7.6.1 多关键码排序	273
6.2.1 顺序查找	223	7.6.2 链式基数排序	273
6.2.2 顺序存储的有序表查找	224	7.7 排序方法比较	275
6.3 树结构查找	227	7.7.1 性能比较	275
6.3.1 二叉排序树	227	7.7.2 不同排序方法的适用情况	276
6.3.2 平衡二叉树	234	7.8 典型例题	276
6.4 散列表查找	240	7.9 本章小结	278
6.4.1 基本概念	240	7.9.1 知识点小结	278
6.4.2 散列函数的构造方法	241	7.9.2 扩展学习导读	280
6.4.3 处理冲突的方法	243	练习题	280
6.4.4 散列表的性能分析	246	实验题	281
6.5 查找方法的比较	247	题目 各种内部排序的性能比较	281
6.6 典型例题	247	第8章 扩展应用举例	283
		8.1 求最大子段和	283
		8.1.1 问题描述	283

8.1.2	问题分析与解决	283
8.2	表达式树的构造	287
8.2.1	问题描述	287
8.2.2	问题分析与解决	287
8.3	由等价关系求划分	290
8.3.1	问题描述	290
8.3.2	问题分析与解决	290
8.4	本章小结	293
8.4.1	知识点小结	293
8.4.2	扩展学习导读	293
	练习题	294
	实验题	294
题目 1	模拟银行排队办理业务	294

题目 2	0-1 背包问题	295
附录		296
附录 A	实验要求	296
附录 B	模拟试卷	298
模拟试卷一	(本科水平)	298
模拟试卷二	(本科水平)	300
模拟试卷三	(研究生入学考试水平)	302
模拟试卷四	(研究生入学考试水平)	304
附录 C	2012 年全国硕士研究生入学 统一考试计算机科学与技术 学科联考试卷(“数据结构” 部分)	306
参考文献		312

第1章 数据结构与算法

计算机科学是一门研究数据表示和数据处理的科学。数据就是对客观事务采用的计算机能够识别、存储和处理的符号表示。简言之，数据是计算机化的信息，是计算机可以直接处理的最基本和最重要的对象。无论是进行科学计算或数据处理、过程控制，还是对文件的存储和检索等计算机应用，都是对数据进行加工处理的过程。计算机对数据的处理并不是简单地将数据堆积在一起，而是使其具有某种内在的联系。因此，为了更有效地处理数据，设计出好的算法，编写出结构清晰而且效率高的程序，必须研究数据的特性、数据间的相互关系及其对应的存储表示，并利用这些特性和关系设计出相应的算法和程序。

1.1 引言

数据结构是计算机科学与技术专业的专业基础课，是十分重要的核心课程。数据结构的知识为后续专业课程的学习提供必要的知识和技能准备，学好“数据结构”这门课程，对于学习计算机专业的其他课程，如操作系统、编译原理、数据库管理系统、软件工程、人工智能等都是十分有益的，而且所有的计算机系统软件和应用软件都要用到各种类型的数据结构。因此，要想更好地运用计算机来解决实际问题，仅掌握几种计算机程序设计语言是难以应付众多复杂的课题的，要想有效地使用计算机、充分发挥计算机的性能，还必须学习和掌握好数据结构的有关知识。

1.1.1 为什么要学习数据结构

在计算机发展的初期，人们使用计算机的目的主要是处理数值计算问题。使用计算机来解决一个具体问题时，一般需要经过下列几个步骤：首先要从该具体问题抽象出一个适当的数学模型，然后设计或选择一个解此数学模型的算法，最后编出程序进行调试、测试，直至得到最终的解答。由于当时所涉及的运算对象是简单的整型、实型或布尔类型数据，数据量小且结构简单，所以程序设计者的主要精力是集中于程序设计的技巧上，而无须重视如何组织数据。

随着计算机应用领域的日益广泛和软、硬件技术的发展，非数值计算问题越来越显得重要。据资料统计，当今处理非数值计算性问题占用了90%以上的机器时间。这类问题涉及的数据结构更为复杂，数据元素之间的相互关系一般无法用数学方程式加以描述。因此，解决这类问题的关键不再是数学分析和计算方法，而是要设计出合适的数据结构，才能有效地解决问题。下面所列举的就是属于这一类的具体问题。

【例1-1】 成绩检索系统。要求成绩检索系统提供自动查询的功能，如查找某个学生的单科成绩或平均成绩，查询某门课程的最高分等。

实现这个系统，首先需要考虑如何组织数据，然后再按照相应的算法编写程序，就可以

实现计算机自动检索。例如，可以将每个学生的各项信息（学号、姓名、各项成绩等）用某种构造的数据类型表示，全部考生的信息按学号的次序排列，可以组织成一个线性表格，如表 1-1 所示，根据查询的需要可以设计出各种查询的算法。

表 1-1 学生成绩表

学号	姓名	考试成绩			平均成绩
		高等数学	C 语言	英语	
20121801	吴承志	90	95	85	90
20121802	李淑芳	88	76	91	85
20121803	刘丽	92	78	82	84
20121804	张会友	81	78	72	77
20121805	石宝国	76	82	79	79
20121806	何文颖	86	90	91	89
20121807	赵胜利	76	78	80	78
20121808	崔文靖	82	93	86	87
20121809	刘丽	80	85	81	82
⋮	⋮	⋮	⋮	⋮	⋮

类似的还有电话自动查号系统、图书信息检索系统、仓库库存管理系统等。在这类文档管理的数学模型中，计算机处理的对象之间通常存在着的是一种简单的线性关系，这类数学模型可称为线性的数据结构。

【例 1-2】 棋盘布局问题。要求将 4 个棋子布在 4 行 4 列的棋盘上，使得任两个棋子既不在同一行或同一列，也不在同一对角线上。

此问题的处理过程不是根据某种确定的计算法则，而是利用试探和回溯的探索技术求解。为了求得合理布局，在计算机中要存储布局的当前状态。从最初的布局状态开始，一步步地进行试探，每试探一步形成一个新的状态，整个试探过程形成了一棵隐含的状态树，如图 1-1 所示。回溯法求解过程实质上就是一个遍历状态树的过程。在这个问题中所出现的树也是一种数据结构，它可以应用在许多非数值计算的问题中。

【例 1-3】 教学计划编排问题。一个教学计划包含许多课程，在教学计划包含的许多课程之间，有些必须按规定的先后次序进行，有些则没有次序要求。也就是说，有些课程之间有先修和后续的关系，有些课程可以任意安排次序，如表 1-2 所示。

表 1-2 计算机专业的课程设置

课程编号	课程名称	先修课程	课程编号	课程名称	先修课程
c ₁	计算机导论	无	c ₆	接口技术	c ₃
c ₂	数据结构	c ₁ , c ₄	c ₇	数据库原理	c ₂ , c ₉
c ₃	汇编语言	c ₁	c ₈	编译原理	c ₄
c ₄	C 语言程序设计	c ₁	c ₉	操作系统	c ₂
c ₅	计算机图形学	c ₂ , c ₃ , c ₄			

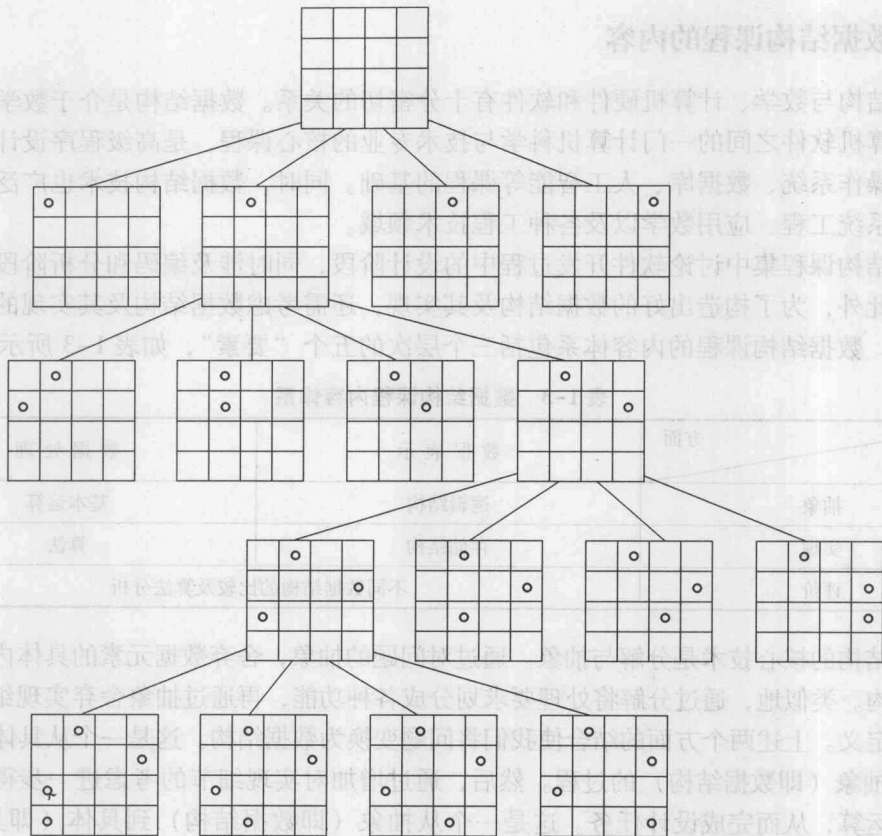


图 1-1 棋盘布局问题中隐含的状态树

计算机专业各门课程之间的次序关系可用一个称做图的数据结构来表示，如图 1-2 所示。有向图中的每个顶点表示一门课程，如果从顶点 c_i 到 c_j 之间存在有向边 $\langle c_i, c_j \rangle$ ，则表示课程 i 必须先于课程 j 进行。

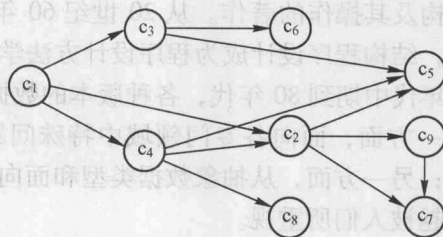


图 1-2 表示课程之间优先关系的有向图

由以上三个例子可见，描述这类非数值计算问题的数学模型不再是数学方程，而是诸如表、树、图之类的数据结构。因此，可以说数据结构课程主要是研究非数值计算的程序设计问题中所出现的计算机操作对象以及它们之间的关系和操作的学科。

学习数据结构的目的是为了了解计算机处理对象的特性，将实际问题中所涉及的处理对象在计算机中表示出来并对它们进行处理。与此同时，通过算法训练来提高学生的逻辑思维能力，通过程序设计的技能训练来提高学生的综合应用能力和专业素质。

1.1.2 数据结构课程的内容

数据结构与数学、计算机硬件和软件有十分密切的关系。数据结构是介于数学、计算机硬件和计算机软件之间的一门计算机科学与技术专业的核心课程，是高级程序设计语言、编译原理、操作系统、数据库、人工智能等课程的基础。同时，数据结构技术也广泛应用于信息科学、系统工程、应用数学以及各种工程技术领域。

数据结构课程集中讨论软件开发过程中的设计阶段，同时涉及编码和分析阶段的若干基本问题。此外，为了构造出好的数据结构及其实现，还需考虑数据结构及其实现的评价与选择。因此，数据结构课程的内容体系包括三个层次的五个“要素”，如表 1-3 所示。

表 1-3 数据结构课程内容体系

层次	方面	数据表示	数据处理
抽象		逻辑结构	基本运算
实现		存储结构	算法
评价		不同数据结构的比较及算法分析	

数据结构的核⼼技术是分解与抽象。通过对问题的抽象，舍弃数据元素的具体内容，就得到逻辑结构。类似地，通过分解将处理要求划分成各种功能，再通过抽象舍弃实现细节，就得到运算的定义。上述两个方面的结合使我们将问题变换为数据结构。这是一个从具体（即具体问题）到抽象（即数据结构）的过程。然后，通过增加对实现细节的考虑进一步得到存储结构和实现运算，从而完成设计任务。这是一个从抽象（即数据结构）到具体（即具体实现）的过程。熟练地掌握这两个过程是数据结构课程在专业技能培养方面的基本目标。

数据结构作为一门独立的课程在国外是从 1968 年才开始的，但在此之前其有关内容已散见于编译原理及操作系统之中。20 世纪 60 年代中期，美国的一些大学开始设立有关课程，但当时的课程名称并不叫数据结构。1968 年美国唐·欧·克努特教授开创了数据结构的最初体系，他所著的《计算机程序设计技巧》第一卷《基本算法》是第一本较系统地阐述数据的逻辑结构和存储结构及其操作的著作。从 20 世纪 60 年代末到 70 年代初，出现了大型程序，软件也相对独立，结构程序设计成为程序设计方法学的主要内容，人们越来越重视数据结构。从 20 世纪 70 年代中期到 80 年代，各种版本的数据结构著作相继出现。目前，数据结构的发展并未终结，一方面，面向各专门领域中特殊问题的数据结构得到研究和发 展，如多维图形数据结构等；另一方面，从抽象数据类型和面向对象的观点来讨论数据结构已成为一种新的趋势，越来越被人们所重视。

1.2 数据结构的概念

在系统地学习数据结构知识之前，需要先了解一些概念和术语。

1.2.1 基本概念和术语

1. 数据 (Data)

数据是信息的载体，是所有能够被计算机识别、存储和加工处理的符号的总称。它是计

计算机程序加工的原料，应用程序可以处理各种各样的数据。在计算机科学中，数据是计算机加工处理的对象，它可以是数值数据，也可以是非数值数据。数值数据是一些整数、实数和复数，主要用于工程计算、科学计算和商务处理等；非数值数据包括字符、文字、图形、图像、语音等。

2. 数据项 (Data Item)

数据项是具有独立含义的标识单位，是数据不可分割的最小单位，如学生成绩表中的“学号”、“姓名”等。数据项有名和值之分，数据项名是一个数据项的标识，用变量定义，而数据项值是它的一个可能取值，表 1-1 中“20121801”是数据项“学号”的一个取值。数据项具有一定的类型，依数据项的取值类型而定。

3. 数据元素 (Data Element)

数据元素是数据的基本单位。在不同的条件下，数据元素又可称为元素、结点、顶点、记录等。例如，考试查分系统的成绩表中的一个记录、棋盘布局问题中状态树的一个状态、教学计划编排问题中的一个顶点等，都被称为一个数据元素。

有时，一个数据元素可由若干数据项组成。例如，学籍管理系统中学生信息表的每一个数据元素就是一个学生记录。它包括学生的学号、姓名、性别、籍贯、出生年月、成绩等数据项。这些数据项可以分为两种：一种叫做初等项，如学生的性别、籍贯等，这些数据项是在数据处理时不能再分割的最小单位；另一种叫做组合项，如学生的成绩，它可以再划分为数学、物理、化学等更小的项。通常，在解决实际应用问题时是把每个学生记录当做一个基本单位进行访问和处理的。

4. 数据对象 (Data Object)

数据对象或数据元素类 (Data Element Class) 是具有相同性质的数据元素的集合。在某个具体问题中，数据元素都具有相同的性质 (元素值不一定相等)，属于同一数据对象 (数据元素类)，数据元素是数据元素类的一个实例。例如，在交通咨询系统的交通网中，所有的顶点是一个数据元素类，顶点 A 和顶点 B 各自代表一个城市，是该数据元素类中的两个实例，其数据元素的值分别为 A 和 B。

5. 数据结构 (Data Structure)

数据结构是指互相之间存在着一种或多种关系的数据元素的集合。数据结构涉及数据元素之间的逻辑关系，数据在计算机中的存储方式和这些数据上定义的一组运算，一般称这三个方面为数据的逻辑结构、数据的存储结构和数据的运算。

(1) 逻辑结构

在任何问题中，数据元素之间都不会是孤立的，在它们之间都存在着这样或那样的逻辑关系，这种数据元素之间的关系称为逻辑结构。数据的逻辑结构包含两个要素：一个是数据元素的集合；另一个是关系的集合。

在形式上，数据的逻辑结构通常可以采用一个二元组来表示：

$$\text{Data_Structure} = (D, R)$$

其中，D 是数据元素的有限集，R 是 D 上关系的有限集。

根据数据元素间关系的不同特性，数据的逻辑结构通常分为以下四类：

1) 集合：在集合中，数据元素间的关系是“属于同一个集合”。集合是元素关系极为松散的一种结构。

- 2) 线性结构：该结构的数据元素之间存在着一对一的关系。
- 3) 树结构：该结构的数据元素之间存在着一对多的关系。
- 4) 图结构：该结构的数据元素之间存在着多对多的关系，图结构也称做网状结构。图 1-3 所示为四类逻辑结构的示意图。

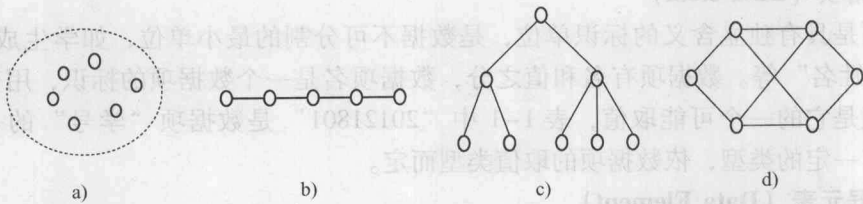


图 1-3 四类基本结构的示意图

a) 集合结构 b) 线性结构 c) 树结构 d) 图结构

由于集合是数据元素之间关系极为松散的一种结构，因此也可用其他结构来表示它。

数据的逻辑结构可以被看做是从具体问题抽象出来的数学模型，它与数据的存储无关。我们研究数据结构的目的是为了在计算机中实现对它的操作，为此还需要研究如何在计算机中表示一个数据结构。数据结构在计算机中的表示（又称映像）称为数据的物理结构，又称存储结构。它所研究的是数据的逻辑结构在计算机中的实现方法，包括数据结构中元素的表示及元素间关系的表示。

(2) 存储结构

数据的存储结构最常用的是顺序存储和链式存储的方法。

1) 顺序存储方法：是把逻辑上相邻的元素存储在物理位置相邻的存储单元中，结点间的逻辑关系由存储单元的邻接关系来体现，由此得到的存储表示称为顺序存储结构。顺序存储结构是一种最基本的存储表示方法，通常借助于程序设计语言中的数组来实现。

2) 链式存储方法：对逻辑上相邻的元素不要求其物理位置相邻，元素间的逻辑关系通过相关联的指示来表示，由此得到的存储表示称为链式存储结构，链式存储结构通常借助于程序设计语言中的指针类型或引用类型来实现。

除了顺序存储方法和链式存储方法外，有时为了查找的方便还采用索引存储方法和散列存储方法。

3) 索引存储方法：是在存储结点信息的同时，还建立附加的索引表。索引表中的每一项包含关键字和地址，关键字是能够唯一标识一个数据元素的数据项，地址指示出数据元素所在的存储位置。索引存储主要是针对数据内容的存储，而不强调关系的存储，索引存储方法主要面向查找操作。

4) 散列存储方法：是以数据元素的关键字的值为自变量，通过某个函数（散列函数）计算出该元素的存储位置。索引存储也是针对数据内容的存储方式。

以上四种存储方法中，顺序存储方法和链式存储方法是最基本与最常用的，索引存储方法和散列存储方法在具体实现时需要用到前两种方法。在实际应用中，一种逻辑结构可以有不同的存储方法，选用何种存储结构来表示相应的逻辑结构要视具体情况而定，主要考虑运算的实现及算法的时空要求。

(3) 数据的运算

运算是^①对数据的处理。运算与逻辑结构紧密相连，每种逻辑结构都有一个运算的集合。运算的种类很多，根据操作的结果，可将运算分为两种类型：

1) 引用型运算：这类运算不改变数据结构中原有的数据元素的状态，只根据需要读取某些信息。

2) 加工型运算：这类运算的结果会改变数据结构中原有数据的状态，如数据元素的内容、个数等。

数据的运算是定义在数据的逻辑结构上的，但运算的具体实现是在数据的存储结构上进行的。数据的运算是数据结构不可分割的一个方面，在数据的逻辑结构和存储结构给定之后，如果定义的运算集及运算的性质不同，也会导致完全不同的数据结构，例如后面章节中将要介绍的线性表、栈和队列等。

1.2.2 抽象数据类型

1. 数据类型 (Data Type)

数据类型是和数据结构密切相关的一个概念。它最早出现在高级程序设计语言中，用以刻画程序中操作对象的特性。在用高级语言编写的程序中，每个变量、常量或表达式都有一个它所属的确定的数据类型。类型显式地或隐含地规定了在程序执行期间变量或表达式所有可能的取值范围，以及在^②这些值上允许进行的操作。因此，数据类型是一个值的集合和定义在这个值集上的一组操作的总称。

在高级程序设计语言中，数据类型可分为两类：一类是原子类型；另一类则是结构类型。原子类型的值是^③不可分解的。如 Java 语言中的整型、字符型、浮点型、双精度型、布尔型等基本类型，分别用保留字 int、char、float、double、boolean 标识。而结构类型的值是由若干成分按某种结构组成的，因此是可分解的，并且它的成分可以是非结构的，也可以是结构的。例如，数组的值由若干分量组成，每个分量可以是整数，也可以是数组等。在某种意义上，数据结构可以被看成是“一组具有相同结构的值”，而数据类型则可被看成是由一种数据结构和定义在其上的一组操作所组成的。

2. 抽象数据类型 (Abstract Data Type, ADT)

抽象数据类型是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关。也就是说，不论其内部结构如何变化，只要它的数学特性不变，都不影响其外部的使用。

抽象数据类型和数据类型实质上是一个概念。例如，各种计算机都拥有的整数类型就是一个抽象数据类型，尽管它们在不同处理器上占用的字节数不尽相同，整数运算的实现方法也可以不同，但由于其定义的数学特性相同，在用户看来都是相同的。因此，“抽象”的意义在于数据类型的数学抽象特性。

但在另一方面，抽象数据类型的范畴更广，它不再局限于前述各处理器中已定义并实现的数据类型，还包括用户在设计软件系统时自己定义的数据类型。为了提高软件的重用性，在近代程序设计方法学中，要求在构成软件系统的每个相对独立的模块上，定义一组数据和施于这些数据上的一组操作，并在模块的内部给出这些数据的表示及其操作的细节，而在模块的外部使用的只是抽象的数据及抽象的操作。这也就是面向对象的程序设计方法。

抽象数据类型的定义可以由一种数据结构和定义在其上的一组操作组成，而数据结构又包括数据元素及元素间的关系，因此抽象数据类型一般可以由元素、关系及操作三种要素来定义。

抽象数据类型的特征是使用与实现相分离，实行封装和信息隐蔽。也就是说，在抽象数据类型设计时，把类型的定义与其实现分离开来。

根据抽象数据类型的构成，可采用如下的形式进行定义。

抽象数据类型名

{ 数据对象: <数据对象集合 D 的定义>

数据关系: <数据对象关系的集合 R 的定义>

基本操作:

<操作列表>

}

【例 1-4】 抽象数据类型“矩阵”的定义。

ADT Matrix

{ 数据对象: $D = \{ a_{ij} \mid i = 1, 2, 3, \dots, m; j = 1, 2, 3, \dots, n; a_{ij}$ 为矩阵中的元素, m 和 n 为矩阵的行数和列数 }

数据关系: $R = \{ \text{Row}, \text{Col} \}$

$\text{Row} = \{ \langle a_{ij}, a_{ij+1} \rangle \mid 1 \leq i \leq m, 1 \leq j \leq n-1 \}$

$\text{Col} = \{ \langle a_{ij}, a_{i+1j} \rangle \mid 1 \leq i \leq m-1, 1 \leq j \leq n \}$

基本操作:

CreateMatrix(M); //创建矩阵

DeleteMatrix(M); //删除矩阵

TransposeMatrix(M); //求矩阵转置

AddMatrix(M, N); //求矩阵 M 和矩阵 N 之和

MulMatrix(M, N); //求矩阵 M 和矩阵 N 之积

} ADT Matrix

1.3 数据结构的描述方法

根据数据结构的概念，在计算机中表示用户自定义的数据结构，实现数据结构上的操作及应用，需要描述三方面的内容：数据对象的类型、数据对象的关系和对数据对象的操作。下面介绍常用的三种描述数据结构的描述方法：C 语言描述、C++ 语言描述和 Java 语言描述。本书采用的数据结构描述方法是完全的面向对象语言——Java 语言描述。读者可参考本节内容，很方便地将书中的算法描述转换为 C 语言描述或 C++ 语言描述。

1.3.1 数据结构的 C 语言描述

C 语言不是面向对象的程序设计语言，因此不具有将数据结构三方面的内容封装的功能，必须分别描述。

数据对象的类型可以是 C 语言提供的 int、char、float、double 等基本数据类型，也可以是用户自定义的数组、结构体、共用体等数据类型。为了表示方便，将数据对象的类型抽象地表