

高等院校信息技术规划教材

数据结构 (第2版)

闫玉宝 等 编著

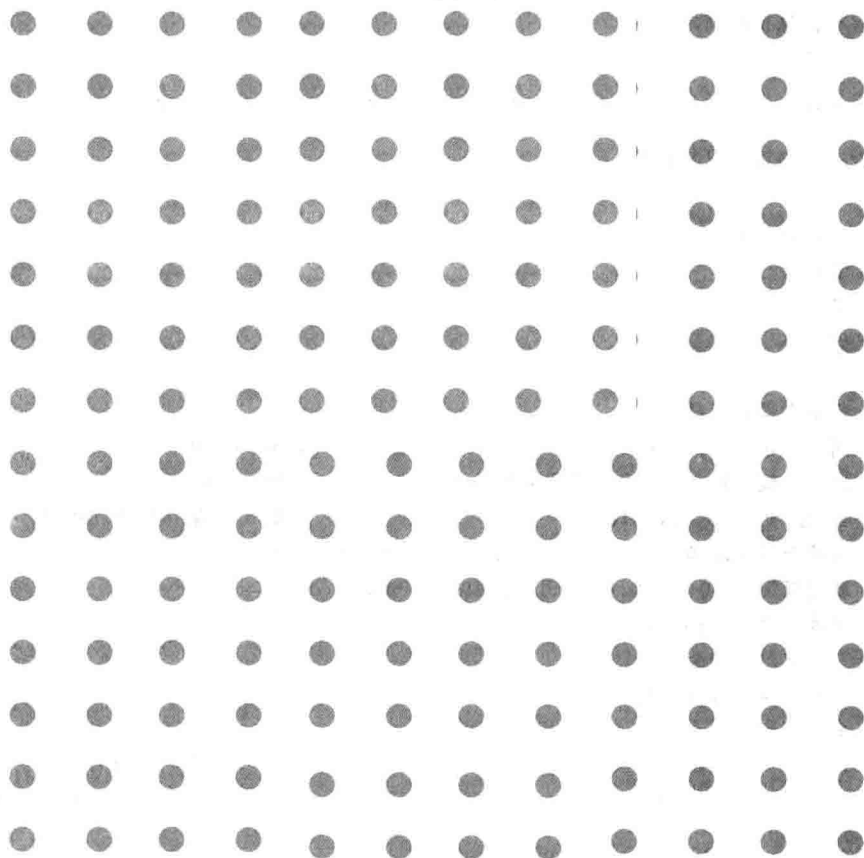


清华大学出版社

高等院校信息技术规划教材

数据结构（第2版）

闫玉宝 李宁 胡超 游静 徐守坤 编著



清华大学出版社
北京

内 容 简 介

本书的特点是简明、易懂、逻辑性强。内容涵盖了线性表、栈、队列、串、数组和广义表、二叉树、树、图以及查找、排序等基础知识。书中提供了适量的数据结构应用实例,让读者在学习数据结构时逐步了解数据结构的应用。各章都配有适量习题,题型多样,难度适中,既适合课堂教学,又便于读者自学时对基础知识的理解和掌握。对于存储结构和算法都给出了完整的C语言实现,描述形式多样化;所有的算法和实例程序都在VC++ 6.0的环境下编译通过,并能正确运行。本书要求读者熟悉C语言。

本书适合作为高等学校计算机和软件工程等专业“数据结构”课程教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构/闫宝玉等编著. —2版. —北京:清华大学出版社,2014

高等院校信息技术规划教材

ISBN 978-7-302-35290-7

I. ①数… II. ①闫… III. ①数据结构—高等学校—教材 IV. ①TP311.12

中国版本图书馆CIP数据核字(2014)第018843号

责任编辑:袁勤勇 顾 冰

封面设计:傅瑞学

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:三河市君旺印装厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.5 字 数:486千字

版 次:2008年3月第1版 2014年3月第2版 印 次:2014年3月第1次印刷

印 数:1~2000

定 价:34.50元

产品编号:053601-01

编委会名单

主 任： 朱 敏

副主任： 王正洪 鲁宇红 焦金生

成 员： （按拼音排序）

常晋义	邓 凯	范新南	高佳琴	高玉寰	龚运新
顾建业	顾金海	林 罡	刘训非	马正华	沈孟涛
唐 全	王继水	王 骏	王 晴	王志立	吴访升
肖 玉	杨长春	袁启昌	张旭翔	张 燕	赵明生
郑保平	郑成增	周凤石			

策划编辑： 袁勤勇

序

preface

在科教兴国方针的指引下,我国高等教育进入了一个新的历史发展时期,招生规模 and 在校生数量都有了大幅度的增长。我们在进行着世界上规模最大的高等教育。与此同时,对于高等教育的研究和认识也在不断深化。高等学校要明确自己的办学方向和办学特色,这既是不断提高高等教育水平的必然要求,更是高校不断发展和壮大必须首先考虑的问题。

教育部领导明确提出,高等教育应多元化,高等院校应实施分类、分层次教学,这是高等教育大众化的必然结果,也是市场对人才需求的客观规律所致。因此要有相当部分的高等院校致力于培养应用型人才。此类院校在计算机教学中如何实现自己的培养目标,如何选择适用的应用型教材,已成为十分重要和迫切的任务。应用型人才的培养不能简单照搬研究型人才的培养模式,要在丰富的实践基础上认真总结,摸索新形势下的教学规律,在此基础上设计相关课程,改进教学方法,同时编写与之相适应的应用型教材。这一工作是非常艰巨的,也是非常有意义的。

在清华大学出版社的大力支持和配合下,于2003年成立了应用型教材编委会。编委会汇集了众多高等院校的实践经验,并经过集中讨论和专家评审,遴选了一批优秀教材,希望能够通过这套教材的出版和使用,促进应用型人才培养的实践发展,为建立新的人才培养模式作出贡献。

我们编写应用型教材的主要出发点是:

1. 适应新形势下教育部“质量工程”对高等教育的要求,以及市场对应用型人才的需求。
2. 计算机科学技术和信息技术发展迅速,教材内容和教学方式应与之相适应,适时地进行更新和改进。
3. 教育技术的发展对教材建设提出了更高的要求,教材将呈现

出纸介质出版物、电子课件以及网络学习环境等相互配合的立体化形态。

4. 根据不同的专业要求,突出应用,使理论与实践更加紧密结合。

以此为目标,我们将努力编写一套全新的、有实用价值的应用型计算机教材。经过参编教师的努力,第一批教材已经面世。教材将滚动式地不断更新、修正、提高,逐渐树立起自己的品牌。希望使用本系列教材的广大师生能对我们的教材提出宝贵的意见,共同建设具有应用型特色的精品教材。

朱 敏

2006 年 5 月

前言

foreword

“数据结构”是计算机和软件工程等专业基础核心课程之一。本书系统地介绍了软件设计中常用的数据结构及其存储结构和实现算法,以及常用的查找和排序算法,并对其进行性能分析和比较。本书内容丰富,简明实用。

本书在保持第1版的“以基本知识点的掌握为基础,实际应用培养为重点”的原则上进行了较大的修改。修改了第1版中存在的错误和不足;进行了章节调整,把树和二叉树分成两章进行介绍;对部分章节的实例、习题进行了增删;算法和设计实例都采用源码的形式描述,目的在于减少读者阅读理解的难度。十分注重与C程序设计课程的衔接,在讲授数据结构的同时,可以进一步加强读者对程序设计的理解。描述的结构形式多样,所有的算法都有完整的C程序实现,并都在VC++ 6.0环境下编译通过,运行正确。

本书共11章,第1章是绪论,介绍什么是数据结构、什么是算法以及算法复杂度分析的方法。第2~6章分别介绍各种线性结构及其基本操作,包括线性表、栈、队列、串以及数组和广义表;第7、第8章介绍树形结构及其基本操作,包括二叉树和树;第9章介绍图形结构及其基本操作;第10、第11章分别介绍常用的查找和排序算法,并对其进行性能分析。每一章都选配了适量的实例和习题。

本书在编写过程中,参考了一些国内外优秀的相关教材,再次向这些教材的作者表示衷心的感谢。由于编者水平有限,错误和不足之处在所难免,敬请同行及读者批评指正。

作 者

2013年11月7日

目录

Contents

第 1 章 绪论	1
1.1 数据结构	1
1.1.1 基本概念	1
1.1.2 数据的逻辑结构	2
1.1.3 数据的存储结构	3
1.1.4 数据结构的操作	3
1.1.5 数据结构研究的内容及作用	5
1.2 算法	6
1.2.1 什么是算法	6
1.2.2 算法的描述	7
1.2.3 算法设计的目标	7
1.2.4 算法效率分析	8
1.2.5 算法存储空间分析	10
1.3 数据结构、算法和程序的关系	10
1.3.1 数据结构与算法	10
1.3.2 数据结构与程序	11
1.4 算法效率的典型例题	12
1.5 本章小结	15
1.6 习题	15
第 2 章 线性表	18
2.1 线性表的逻辑结构	18
2.1.1 线性表的定义	18
2.1.2 线性表的基本操作	19
2.2 线性表的顺序存储结构	20
2.2.1 顺序表	20
2.2.2 顺序表的基本运算	23

2.3	顺序表的查找	25
2.3.1	按位置查找元素	25
2.3.2	按内容查找元素	25
2.3.3	顺序表的查找操作的效率分析	27
2.4	顺序表的插入与删除	27
2.4.1	在顺序表的第 i 个位置插入一个元素	27
2.4.2	删除顺序表的第 i 个位置元素	28
2.4.3	顺序表的插入与删除操作的效率分析	29
2.5	顺序表的典型例题	30
2.6	线性表的链式存储结构	32
2.6.1	单链表	32
2.6.2	循环链表	34
2.6.3	双向链表	35
2.6.4	静态链表	36
2.7	单链表的建立及其实现	37
2.7.1	创建带头结点的空单链表的算法实现	37
2.7.2	用头插法单链表的插入算法实现	37
2.7.3	用尾插法单链表的插入算法实现	38
2.7.4	在第 i 个位置插入结点的单链表插入算法实现	39
2.8	单链表基本运算的实现	41
2.8.1	单链表辅助运算的实现	41
2.8.2	单链表求表长的实现	42
2.8.3	单链表查找操作的实现	42
2.8.4	单链表删除操作的实现	43
2.9	双向链表基本运算的实现	44
2.9.1	双向链表插入操作的实现	45
2.9.2	双向链表删除操作的实现	46
2.10	链表的典型例题	47
2.11	本章小结	50
2.12	习题	50
第3章	栈	53
3.1	栈的逻辑结构	53
3.1.1	栈的定义	53
3.1.2	栈的基本运算	53
3.2	栈的顺序存储与操作实现	54
3.2.1	栈的顺序存储	54
3.2.2	顺序栈的操作实现	55

3.3	栈的链式存储与操作实现	57
3.3.1	栈的链式存储	57
3.3.2	链栈的操作实现	58
3.4	栈的典型举例	60
3.5	本章小结	65
3.6	习题	65
第4章	队列	67
4.1	队列的逻辑结构	67
4.1.1	队列的定义	67
4.1.2	队列的基本操作	68
4.2	队列的顺序存储与操作实现	68
4.2.1	队列的顺序存储表示	68
4.2.2	队列的基本运算	69
4.3	循环队列的存储与操作实现	71
4.3.1	循环队列的顺序存储	71
4.3.2	循环队列的操作实现	72
4.4	队列的链式存储与操作实现	74
4.4.1	队列的链式存储	74
4.4.2	链队列的操作实现	75
4.5	队列的典型举例	77
4.6	本章小结	84
4.7	习题	85
第5章	串	86
5.1	串的定义、表示和实现	86
5.1.1	串的基本概念	86
5.1.2	串的基本操作	87
5.2	串的存储结构	89
5.2.1	串的顺序存储结构	89
5.2.2	串的堆存储结构	91
5.2.3	串的链式存储结构	92
5.3	串的模式匹配	93
5.3.1	简单模式匹配算法	93
5.3.2	改进的模式匹配算法	94
5.4	串的典型例题	98
5.5	算法设计举例	102

5.6	本章小结	108
5.7	习题	108
第6章	数组和广义表	110
6.1	数组	110
6.1.1	数组的定义	110
6.1.2	数组的顺序存储结构	110
6.2	特殊矩阵的压缩存储	112
6.2.1	对称矩阵	112
6.2.2	三角矩阵	113
6.2.3	对角矩阵	114
6.3	稀疏矩阵的压缩存储	115
6.3.1	稀疏矩阵的定义	115
6.3.2	三元组表的存储方法	116
6.3.3	十字链表的存储方法	120
6.4	广义表	121
6.4.1	广义表的概念和基本操作	122
6.4.2	广义表的存储结构	123
6.4.3	广义表基本操作的实现	125
6.5	本章小结	128
6.6	习题	129
第7章	二叉树	132
7.1	二叉树的定义、性质和基本操作	132
7.1.1	二叉树的定义	132
7.1.2	二叉树的基本性质	134
7.1.3	二叉树的基本操作	135
7.2	二叉树的存储	135
7.2.1	顺序存储结构	135
7.2.2	链式存储结构	136
7.3	二叉树的遍历	139
7.3.1	二叉树遍历的方法和结构	139
7.3.2	二叉链表存储结构下二叉树遍历算法的实现	140
7.3.3	非递归的二叉树遍历算法	143
7.3.4	二叉树遍历的应用	145
7.4	线索二叉树	149
7.4.1	二叉树的线索链表	149

7.4.2	以中序线索链表为存储结构的中序遍历	151
7.4.3	线索链表的生成	151
7.5	哈夫曼树及其应用	152
7.5.1	哈夫曼树的定义	152
7.5.2	哈夫曼树的构造方法	153
7.5.3	最优前缀编码	154
7.5.4	哈夫曼编码算法的实现	156
7.6	本章小结	157
7.7	习题	158
第 8 章	树	161
8.1	树的定义与术语	161
8.2	树的存储结构	162
8.2.1	双亲表示法	162
8.2.2	孩子表示法	163
8.2.3	孩子兄弟表示法	164
8.3	树、森林与二叉树的相互转换	166
8.3.1	树与二叉树的相互转换	166
8.3.2	森林与二叉树的相互转换	166
8.3.3	树和森林的遍历	168
8.4	树的典型例题	169
8.5	算法设计举例	172
8.5.1	设计哈夫曼编码	172
8.5.2	前缀算术表达式转换	174
8.6	本章小结	176
8.7	习题	176
第 9 章	图	178
9.1	图的定义	178
9.2	图的基本术语	180
9.3	图的基本操作	182
9.4	图的存储结构	183
9.4.1	邻接矩阵	183
9.4.2	邻接表	186
9.5	图的遍历	188
9.5.1	深度优先遍历	188
9.5.2	广度优先遍历	189

9.6	最小生成树	191
9.6.1	最小生成树的基本概念	191
9.6.2	普里姆算法	192
9.6.3	克鲁斯卡尔算法	194
9.7	最短路径	195
9.7.1	求某一个顶点到其余各顶点的最短路径	195
9.7.2	求任意一对顶点之间的最短路径	198
9.8	关键路径	201
9.8.1	AOV 网与拓扑排序	201
9.8.2	AOE 网与关键路径	204
9.9	图的典型例题	210
9.10	本章小结	218
9.11	习题	219
第 10 章	查找	222
10.1	查找表	222
10.2	静态查找表	222
10.2.1	顺序查找	223
10.2.2	有序表的折半查找	224
10.2.3	分块查找	227
10.3	动态查找表——二叉排序树	227
10.3.1	二叉排序树的查找过程	228
10.3.2	二叉排序树的插入操作	229
10.3.3	二叉排序树的删除操作	230
10.3.4	二叉排序树的查找分析	232
10.4	动态查找表——平衡二叉树	233
10.4.1	左单旋转(LL 型)	233
10.4.2	右单旋转(RR 型)	234
10.4.3	先左后右双向旋转(LR 型)	234
10.4.4	先右后左双向旋转(RL 型)	235
10.4.5	二叉平衡树平衡化算法	235
10.5	动态查找表——B-树和 B+ 树	239
10.5.1	B-树的定义	239
10.5.2	B-树的查找	240
10.5.3	在 B-树上插入结点	242
10.5.4	在 B-树上删除结点	244
10.5.5	B+ 树	245
10.6	哈希表	245

10.6.1	哈希表与哈希方法	246
10.6.2	常用的哈希函数	246
10.6.3	处理冲突的方法	248
10.6.4	哈希表的查找分析	250
10.7	查找的典型例题	251
10.8	算法设计举例	253
10.9	本章小结	261
10.10	习题	261
第 11 章	排序	264
11.1	基本概念	264
11.2	插入排序	265
11.2.1	直接插入排序	265
11.2.2	折半插入排序	267
11.2.3	表插入排序	268
11.2.4	希尔排序	271
11.3	交换排序	272
11.3.1	冒泡排序	272
11.3.2	快速排序	274
11.4	选择排序	277
11.4.1	直接选择排序	277
11.4.2	树形选择排序	278
11.4.3	堆排序	280
11.5	二路归并排序	282
11.6	基数排序	284
11.6.1	多关键码排序	285
11.6.2	链式基数排序	286
11.7	排序的典型例题	289
11.8	算法设计举例	290
11.9	本章小结	292
11.10	习题	293
参考文献		296

第1章

chapter 1

绪 论

现实世界中的任何事与物都可以进行数字化,它们在计算机中的表现均为数据。这些数据来源于现实,有实际的意义,而且在计算机中有统一的表示方法。编写解决问题的程序时,不仅要考虑如何存储待处理的数据,而且要根据数据处理的要求设计数据处理的方法。研究数据在计算机中的表示方法、存储方法以及在其上的处理方法即操作,就构成了数据结构的主要内容。本章主要介绍数据结构和算法的基本概念及相关内容。

1.1 数据结构

计算机科学是一门研究数据表示和数据处理的科学。在利用计算机进行数据处理时,实际需要处理的数据元素一般有很多。要提高数据处理效率且节省存储空间,如何组织数据就成了关键问题。而数据结构用来反映一个数据的内部构成,即数据由哪些成分构成,以什么方式构成,是什么结构。为了更好地回答这些问题,下面先对一些基本概念和术语进行介绍,然后给出数据结构的具体定义。

1.1.1 基本概念

在系统地学习数据结构知识之前,先对一些基本概念和术语赋予确切的含义。

1. 数据(data)

数据是外部世界信息的载体。它能够被计算机识别、存储和加工处理,是计算机程序加工的原料。计算机程序处理各种各样的数据,可以是数值数据,如整数、实数或复数,主要用于工程计算、科学计算和商务处理等,也可以是非数值数据,如字符、文字、图形、图像和声音等。

2. 数据元素(data element)和数据项(data item)

数据元素是数据的基本单位,在计算机程序中通常被作为一个整体进行考虑和处理。数据元素有时也被称为元素、结点、顶点和记录等。

一个数据元素可由若干个数据项组成。数据项是不可分割的、含有独立意义的最小

数据单位。数据项有时也称为字段(field)。例如,在数据库信息处理系统中,数据表中的一条记录就是一个数据元素。这条记录中的学生学号、姓名、性别、籍贯、出生年月和成绩等字段就是数据项。

3. 数据类型(data type)

数据类型是高级程序设计语言中的概念,是数据的取值范围和对数据进行操作的总和。数据类型规定了程序中对象的特性。程序中的每个变量、常量或表达式的结果都应该属于某种确定的数据类型。例如,C语言中的整型变量,其取值范围为某个区间上的整数(区间大小因机器而异),定义在其上的操作为加、减、乘、除和取模等算术运算。再如结构体(struct),它是C语言中聚合数据类型的一类,结构体可以被声明为变量、指针或数组等,用于实现较复杂的数据结构。结构体同时也是一些元素的集合,这些元素称为结构体的成员(member),且这些成员可以为不同的类型。

4. 数据结构(data structure)

简单地说,数据结构是指数据与数据之间的关系。在任何问题中,数据元素之间都不是孤立的,都存在着一定的关系,这种关系称为结构(structure)。

在实际使用时,这种关系需要在计算机中进行表示。因此,特别地,将这种数据之间的逻辑关系描述称为数据的逻辑结构,而将其在计算机中的表示(即映像)称为数据的物理结构,又称为存储结构。

1.1.2 数据的逻辑结构

数据的逻辑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。根据数据元素间关系的不同特性,通常有下列4类基本的结构。

(1) 集合结构:在集合结构中,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,即一个数据元素只与另一个数据元素有关系。

(3) 树形结构:该结构的数据元素之间存在着一对多的关系,即一个数据元素与另外多个数据元素有关系。

(4) 图形结构:该结构的数据元素之间存在着多对多的关系,即数据元素之间有多个复杂的交叉关系。图形结构也称作网状结构。

图1-1为表示上述4类基本结构的示意图。

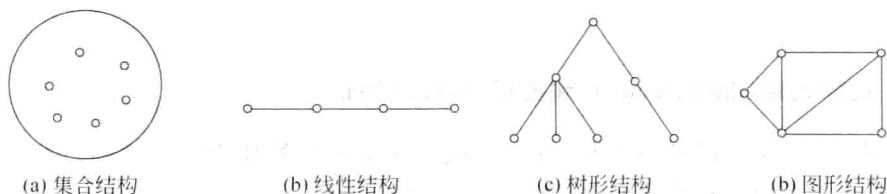


图 1-1 4类基本结构的示意图

从上面所介绍的数据结构的概念中可以知道,一个数据结构有两个要素:一个是数据元素的集合;另一个是关系的集合。在形式上,数据结构通常可以采用一个二元组来表示,形式如下:

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

其中, D 是含有有限个数据元素的集合, R 是 D 上关系的有限个数据元素的集合。

1.1.3 数据的存储结构

数据的存储结构是建立数据元素的机内表示,也称为数据的物理结构。它包括两个部分:元素数据的存储和元素之间关系的存储。有时为方便操作的实现,还会增加一些辅助信息的存储。由于每个数据元素被表示为一个存储结点,所以元素之间的逻辑关系就由存储结点之间的关联方式间接地表示出来。通常存储结点之间的关联有下列4种方式。

(1) 顺序存储实现:所有的存储结点存放在一块连续的存储区域中,结点之间的逻辑关系通过结点的存储位置来体现。顺序存储通常用数组来实现。

(2) 链表存储实现:链表存储中逻辑上相邻的数据元素,存储位置不一定也相邻。把数据元素结点存储信息进行扩展,不仅存储数据元素本身的信息,而且存储数据元素之间逻辑关系的信息。后者就是通过引入指针实现的。

(3) 散列存储实现:此存储中是通过哈希函数对每个结点的关键字计算出存储地址实现的。

(4) 索引存储实现:此存储中除建立存储结点信息外,还建立附加的索引表来标识结点的地址。索引表由若干个索引项组成。索引项一般是关键字和地址等。

1.1.4 数据结构的操作

数据的操作是指作用在某种数据结构上的操作,如查找、插入、删除和修改等操作。数据结构不同,查找、插入、删除以及修改等的操作对象也不同,操作所需的条件以及复杂程度也不同。如在一个线性结构中插入一条记录,只需要知道插入到第几个位置就可以将数据准确地插入到某个位置。线性结构的插入过程如图1-2所示。

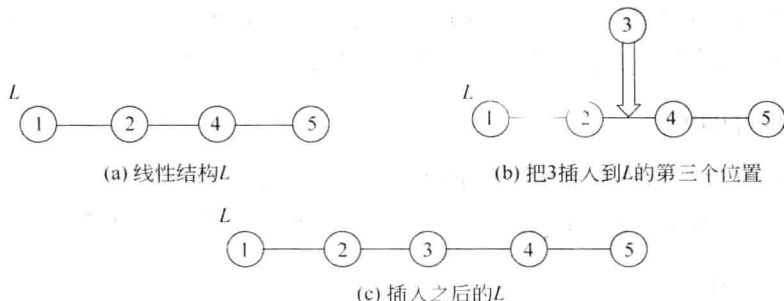


图 1-2 线性结构的插入过程