



普通高等教育“十一五”国家级规划教材

计算机类本科规划教材

数据结构

(第2版)

◎ 田鲁怀 姜吉顺 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十一五”国家级规划教材
计算机类本科规划教材

数 据 结 构

(第2版)

田鲁怀 姜吉顺 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是普通高等教育“十一五”国家级规划教材。全书共 10 章,内容包括:数据结构的概念,几种基本的线性结构(如线性表),栈和队列,串,几种非线性结构(如多维数组和广义表),树,图,常用的数据处理技术(如排序),查找,文件的存储结构和组织方法等。在每章中都收集了难度各异的习题和例题,全书采用 C 语言作为算法描述语言,并有详细的注释,书中全部程序均上机验证并调试通过,同时给出部分程序的运行结果。各章中的“简单应用举例”,既是本章算法的综合应用,也可作为本章实训内容和课程设计的综合练习,全书有很强的实用性和可操作性。

本书可以作为全日制高等学校计算机应用专业、微电子和信息工程专业、计算机信息管理和经济信息管理类专业普通本科学生的专业基础课教材,也可以作为上述专业高职高专学生的参考教材,还可以作为计算机等级考试的参考书,供广大从事计算机应用工作的管理人员和技术人员学习参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

数据结构 / 田鲁怀, 姜吉顺编著. —2 版. —北京: 电子工业出版社, 2016.1

计算机类本科规划教材

ISBN 978-7-121-27693-4

I. ①数… II. ①田… ②姜… 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(2015)第 284168 号

责任编辑: 冉 哲

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 24.25 字数: 620.8 千字

版 次: 2006 年 8 月第 1 版

2016 年 1 月第 2 版

印 次: 2016 年 1 月第 1 次印刷

定 价: 49.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

再 版 前 言

本书是普通高等教育“十一五”国家级规划教材。本书较系统地介绍了程序设计中最常用的一些数据结构,如线性表、栈和队列、数组、串、树、图等;阐述各种数据结构的逻辑关系,讨论它们在计算机中的存储表示,以及在這些数据结构上的运算,并对其算法的复杂度进行简要的分析;另外,还介绍了程序设计中常用的各种排序和查找算法;每章中的“简单应用举例”,既是本章算法的综合应用,也可作为本章实训内容和课程设计的综合练习,具有很强的实用性和可操作性。

本书共分 10 章,内容包括:数据结构的概念,几种基本的线性结构(如线性表),栈和队列,串,几种非线性结构(如多维数组和广义表),树,图,常用的数据处理技术(如排序),查找,文件的存储结构和组织方法等。

本书既重视理论又重视实践,配有大量的习题和例题,解释颇为详细。全书采用 C 语言作为描述语言,并有详细的注释,书中全部程序均上机验证并调试通过,同时给出部分程序的运行结果。

本书是作者在讲授“数据结构”和“高级程序设计语言”等课程 20 余年教学经验的基础上,集历年各种版本教材及作者多年来的备课笔记之精华,结合本科教学特点编写而成的。因此,学生通过相关内容的学习之后,可以直接使用书中算法或程序,有助于提高学生对所学知识的融会贯通和灵活应用,培养学生理论联系实际的良好学风。

本书第 1 版于 2006 年出版,经过近 10 年在上海交通大学、上海大学、海南大学、山东理工大学等多所高校的计算机及应用、微电子和信息工程、信息安全、计算机信息管理和自动控制等相关专业的多届学生中的使用,取得了较好的教学效果并获得了较高的评价。另外,本书还于 2009 年 9 月获得上海交通大学第 12 届优秀教材二等奖。

本书在再版过程中,山东理工大学电气与工程学院姜吉顺副教授对全书进行了全面修订,并对书中的例题和习题以及书中的全部程序进行了验证和调试。

本书的再版不仅更正了原书中的疏漏和错误,也增补了一些细节;另外,在文字的叙述方面也进行了一些润色和删繁就简的工作。

此书的再版,首先要感谢电子工业出版社和冉哲编辑。因为目前的出版业与 10 年前已发生很大的变化,市场的竞争更加激烈、更为市场化。所以,对于出版社来说,决定一本书是否再版,是需要一定的眼光和需要承担一定风险的。但是,对于一个作者来说,能够在这种情况下,再版一本旧作,比完成一本新书更有成就感。因为,书作为科学的载体,其价值是需要用时间来检验的,只有经得起时间考验的作品,才是真正具有价值的。

上海大学计算机学院原院长张吉锋教授、上海交通大学软件学院原副院长侯文永教授对

本书的内容以及实验和课程设计教学环节提出了很多宝贵的意见与建议；上海大学计算机工程与科学学院的曹旻、刘华两位老师，海南大学信息学院信息安全系的周晓谊老师在本书的使用和推介过程中做了大量的工作，并对此书的部分内容提出了一些建设性的修改意见，谨在此一并表示衷心的感谢。

本书主要作为全日制高等学校计算机应用专业、微电子和信息工程专业、计算机信息管理和经济信息管理类专业普通本科学生的专业基础课教材；也可以作为上述专业高职高专学生的参考教材；还可以作为计算机等级考试的参考书，供广大从事计算机应用工作的管理人员和技术人员学习参考。

由于作者水平有限，书中缺点和错误在所难免，希望广大读者批评指正。

作 者
于上海交通大学

目 录

第 1 章 概论	1	第 3 章 栈和队列	66
1.1 概述	1	3.1 栈的基本概念	66
1.2 数据结构的基本概念	4	3.2 栈的存储结构	67
1.2.1 数据结构的基本术语	4	3.2.1 栈的顺序存储结构	67
1.2.2 数据的逻辑结构	6	3.2.2 栈的链接存储结构	68
1.2.3 数据的存储结构	8	3.2.3 栈的两种存储结构的比较	69
1.3 算法性能分析与度量	12	3.2.4 多个顺序栈共享一个数组的 存储空间	69
1.3.1 算法和算法的描述方法	12	3.3 栈的基本运算	70
1.3.2 算法的特性	14	3.3.1 顺序存储结构上顺序栈的 运算实现	71
1.3.3 算法设计的要求	14	3.3.2 链接存储结构上链栈的运算 实现	72
1.3.4 算法时间复杂度的分析与度量	15	3.4 栈的简单应用举例	73
1.3.5 算法存储空间的分析与度量	19	3.4.1 栈在递归过程中的作用	73
本章小结	19	3.4.2 栈的几个简单应用实例	76
习题 1	20	3.5 队列的基本概念	81
第 2 章 线性表	23	3.6 队列的存储结构	82
2.1 线性表的定义及基本运算	23	3.6.1 队列的顺序存储结构	82
2.1.1 线性表的定义	23	3.6.2 顺序存储的循环队列	84
2.1.2 线性表的基本运算	24	3.6.3 队列的链接存储结构	85
2.2 线性表的顺序存储结构及其运算	25	3.7 队列的基本运算	86
2.2.1 线性表的顺序存储结构	25	3.7.1 顺序存储结构上顺序队列的运算 实现	86
2.2.2 顺序表上的基本运算	26	3.7.2 顺序存储结构上循环队列的运算 实现	87
2.2.3 顺序表上插入和删除运算的 时间分析	30	3.7.3 链接存储结构上链队列的运算 实现	89
2.2.4 顺序表的优点和缺点	31	3.8 队列的简单应用举例	91
2.3 线性表的链接存储结构及其运算	31	本章小结	97
2.3.1 单链表	31	习题 3	98
2.3.2 单链表上的基本运算	32	第 4 章 串	100
2.3.3 单链表上查找、插入和删除 运算的时间分析	40	4.1 串的基本概念	100
2.3.4 循环链表	40	4.2 串的存储结构	101
2.3.5 双向链表	43	4.2.1 串的顺序存储结构	101
2.4 顺序表和链表的比较	46		
2.5 线性表的简单应用举例	47		
本章小结	62		
习题 2	63		

4.2.2 串的链接存储结构	103	6.3.1 二叉树的遍历	180
4.3 串的基本运算及实现	105	6.3.2 二叉树的建立	185
4.3.1 串的基本运算	105	6.3.3 二叉树的其他运算举例	187
4.3.2 顺序串上基本运算的实现	106	6.4 线索二叉树	192
4.3.3 链串上基本运算的实现	108	6.4.1 线索二叉树的概念	192
4.4 串的模式匹配运算	112	6.4.2 二叉树的中序线索化	193
4.4.1 BF 模式匹配算法	112	6.4.3 线索二叉树的遍历和插入运算	195
4.4.2 BM 模式匹配算法	115	6.5 树和森林	198
4.4.3 KMP 模式匹配算法	117	6.5.1 树的存储结构	198
4.5 串的简单应用举例	124	6.5.2 树和森林与二叉树的转换	201
本章小结	131	6.5.3 树的遍历	205
习题 4	132	6.5.4 森林的遍历	206
第 5 章 数组和广义表	133	6.6 哈夫曼树及其应用	207
5.1 数组的概念和存储	133	6.6.1 哈夫曼树的基本概念	207
5.1.1 数组的概念	133	6.6.2 哈夫曼树的构造及实现	208
5.1.2 数组的存储结构	134	6.6.3 哈夫曼编码	211
5.2 特殊矩阵的压缩存储	137	6.6.4 哈夫曼译码	215
5.2.1 对称矩阵的压缩存储	137	6.6.5 哈夫曼树在编码问题中的完整程序	216
5.2.2 三角矩阵的压缩存储	138	本章小结	218
5.2.3 对角矩阵的压缩存储	139	习题 6	219
5.3 稀疏矩阵的压缩存储	141	第 7 章 图	222
5.3.1 稀疏矩阵的三元组表示	141	7.1 图的基本概念	222
5.3.2 稀疏矩阵的十字链表表示	148	7.1.1 图的实际背景	222
5.3.3 稀疏矩阵的简单应用举例	152	7.1.2 图的定义	223
5.4 广义表	157	7.1.3 图的基本术语	224
5.4.1 广义表的基本概念	157	7.2 图的存储结构	227
5.4.2 广义表的链接存储结构	158	7.2.1 邻接矩阵表示法	227
5.4.3 广义表的基本运算	161	7.2.2 邻接表表示法	231
5.4.4 广义表的简单应用举例	166	7.3 图的遍历	234
本章小结	167	7.3.1 连通图的深度优先搜索遍历	235
习题 5	168	7.3.2 连通图的广度优先搜索遍历	237
第 6 章 树	170	7.3.3 非连通图的遍历	240
6.1 树的基本概念	170	7.3.4 连通图和非连通图的建立与遍历运算实例	241
6.1.1 树的定义	170	7.4 生成树和最小生成树	243
6.1.2 树的基本术语	172	7.4.1 生成树和最小生成树的概念	244
6.2 二叉树	174	7.4.2 Kruskal 算法	245
6.2.1 二叉树的概念	174	7.4.3 Prim 算法	248
6.2.2 二叉树的基本性质	176	7.5 最短路径	250
6.2.3 二叉树的存储结构	177		
6.3 二叉树的运算	180		

7.5.1 最短路径的概念	250
7.5.2 单源最短路径	252
7.5.3 所有顶点对之间的最短路径	255
7.6 AOV 网和拓扑排序	260
7.6.1 AOV 网和拓扑排序的概念	260
7.6.2 拓扑排序算法	261
7.7 AOE 网和关键路径	265
7.7.1 AOE 网和关键路径的概念	265
7.7.2 关键路径的确定	267
7.8 图的简单应用举例	269
本章小结	277
习题 7	278
第 8 章 排序	281
8.1 排序的基本概念	281
8.2 插入排序	284
8.2.1 直接插入排序	284
8.2.2 希尔排序	286
8.3 交换排序	288
8.3.1 冒泡排序	288
8.3.2 快速排序	291
8.4 选择排序	294
8.4.1 直接选择排序	294
8.4.2 堆排序	295
8.5 归并排序	302
8.5.1 两个相邻有序表的一次归并过程	303
8.5.2 一趟归并排序过程	303
8.5.3 二路归并排序	304
8.6 各种内排序方法的比较和选择	305
8.6.1 各种内排序方法的总结	305
8.6.2 各种内排序方法的比较	305
8.6.3 排序方法的选择	306
8.7 排序的简单应用举例	307
本章小结	311

习题 8	312
第 9 章 查找	315
9.1 查找的基本概念	315
9.2 线性表的查找	316
9.2.1 顺序查找	316
9.2.2 二分查找	317
9.2.3 分块查找	320
9.3 树表的查找	323
9.3.1 二叉排序树	323
9.3.2 平衡的二叉排序树	330
9.3.3 B-树	335
9.4 散列表的查找	342
9.4.1 散列表的概念	342
9.4.2 散列函数的构造方法	344
9.4.3 处理冲突的方法	347
9.4.4 散列表的运算	351
9.4.5 散列表的查找及分析	355
9.5 查找的简单应用举例	357
本章小结	362
习题 9	363
第 10 章 文件	365
10.1 文件的基本概念	365
10.2 顺序文件	367
10.3 索引文件	368
10.4 索引顺序文件	370
10.4.1 ISAM 文件	370
10.4.2 VSAM 文件	373
10.5 散列文件	375
10.6 多关键字文件	376
10.6.1 多重表文件	376
10.6.2 倒排文件	377
本章小结	378
习题 10	379
参考文献	380

第1章 概 论



内容提要

“数据结构”是一门随着计算机科学的发展而逐渐形成的新兴学科。随着计算机技术的发展,计算机的应用领域从最初的科学计算发展到人类社会的各个领域,计算机处理的对象也由纯粹的数值发展到字符、表格、图像和声音等各种具有一定结构的数据,这就给程序设计带来一些新的问题。与飞速发展的计算机硬件相比,计算机软件的发展相对缓慢。因此,研究数据结构对设计出一个高性能的算法和高性能软件是至关重要的。

本章将介绍数据结构的基本概念:数据、数据元素、数据结构、数据的逻辑结构和存储结构、数据运算、算法和算法分析等。了解这些概念有助于对以后章节加深理解。

1.1 概述

一般来说,用计算机解决一个实际问题时,大致需要以下几个步骤:首先从具体问题抽象出一个适当的数学模型,然后设计一个解此数学模型的算法,最后编出程序进行调试直至得到最终的解答。寻求数学模型的实质是分析问题,从中提取操作的对象,并找出这些操作对象之间的关系,然后用数学的语言加以描述。然而,随着计算机应用领域的不断扩大,计算机处理的更多的是非数值计算问题,例如,图书资料查询、电话号码的自动管理、交通道路规划、博弈游戏等问题。它们的数学模型一般无法用数学方程式加以描述,因此,解决此类问题的关键已不再是分析数学和计算方法,而是要建立有效的数据结构来进行描述,分析问题中所用到的数据是如何组织的,研究数据之间的关系如何,进而为解决这些问题设计出合适的数据结构。为了增加对数据结构的感性认识,下面举例说明数据结构的概念。

【例 1.1】电话号码自动查询问题。

电话号码查询的最主要的工作是,当给出某个单位名称或某个人的姓名时,能在电话号码表中迅速找到其电话号码,若找不到,则给出该单位或个人的电话号码不存在的信息。此外,当有新用户要加入、旧用户要改号或撤销时,要对电话号码表进行相应的修改。那么,如何组织电话号码表,实现上述查询、插入、删除和修改等操作呢?假设电话号码表的组成见表 1.1。表中各用户的电话号码是随机罗列出来的。若要查找某人或某单位的电话号码,就必须从表的开始依次往后顺序查找。若该用户确实注册,就会找到该用户的电话号码。但是,采用这种方式进行查找,效率是很低的。

为了提高查找效率,可以重新组织电话号码表,将单位和私人电话分开登记。单位电话按行业分类组织,将同行业的电话登录在一起,并建立一个分类索引表(分类简表)和行业分类目录,如图 1.1 所示。而私人电话则按姓氏笔画进行登录,同时建立一个姓氏笔画索引表,如图 1.2 所示。

表 1.1 电话号码随机登记表

编 号	单位名称或个人姓名	电 话 号 码
00001	上海市图书馆	12346666（总机）
00002	上海市动物园	12340176（总机）
00003	上海市大饭店	11223344（总机）
00004	上海市医院	18801234
00005	公交问讯电话	16088160
00006	徐虎报修热线	12345678
00007	王文玲	55551234

部门行业	页号
党政机关	17
教育	32
医疗	53
金融	71
服务	75
娱乐	133
交通	140
商业	144
综合	161

行业分类	页号
高等教育	32
初、中等教育	34
学前教育	40
职业、业余教育	42
特殊教育	46
广播电视、函授	48
专业教育	50
老年大学	51
技能鉴定与考核	52

单位名称	电话号码	页号
大同中学	56780973	34
上海中学	56787919	35
华师大一附中	56784040	36
曹阳中学	56789407	37
交大附中	56781010	38
光明中学	56783588	39
复旦附中	56780560	40
复兴高级中学	56781765	41
南洋模范中学	56785748	42

(a) 分类索引表 (b) 行业分类目录索引表 (c) 单位电话号码登记表

图 1.1 单位电话号码组织构造的示意图

姓氏	页号范围	姓名	电话号码	页号
丁	100-101	王易	12375856	104
卜	101-101	王尔	12377372	104
于	101-102	王散	12378405	105
万	102-102	王锋	12334942	109
马	102-104	王宜平	12338343	111
王	104-115	王金贵	12334490	112
韦	115-115	王祥	12348818	113
牛	115-116	王芳	12349759	113
方	116-119	王文玲	55551234	114

(a) 私人住宅电话索引表 (b) 私人住宅电话登记表

图 1.2 私人住宅电话索引表和电话登记表的示意图

若要查找某单位的电话号码，可以先在图 1.1 (a) 分类索引表中查找部门行业，然后根据索引表再从图 1.1 (b) 行业分类目录索引表中进行查找，最后从图 1.1 (c) 所示的单位电话登记表中查找给定单位的电话号码。例如，查找光明中学的电话号码。首先，在分类索引表中找到教育的开始页号 32，然后在行业分类目录索引表中找到初、中等教育的开始页号 34，最后根据该页号就能够迅速找到光明中学的电话号码。若要查找私人住宅电话，则先从图 1.2 (a) 所示的电话索引表中查找姓氏，然后根据索引表在图 1.2 (b) 所示的电话登记表中查找。例如，查找用户王文玲的电话号码。首先，在私人住宅电话索引表中找到王姓，根据页号范围，在私人住宅电话登记表中，从第 104 页开始依次向后查找，若找到王文玲，则查找结束；

若找到第 115 页还没有找到王文玲，则表明电话登记表中没有王文玲的电话号码。显然，用这种方法进行查找，其查找速度快，效率高。

由于上述电话号码的组织方式不同，进行同样的查找工作，其查找算法不同，查找效率也是不同的。

【例 1.2】无序表的顺序查找和有序表的二分查找。

假设某校选修课成绩登记表和学生情况登记表分别参见表 1.2 和表 1.3。在表 1.2 中，学生记录的排列顺序是没有规律的，因此称为无序表。在表 1.3 中，每个学生记录按学号从小到大顺序排列，因此称为有序表。

表 1.2 2001 年第一学期计算机基础选修课成绩登记表

学 号	姓 名	性 别	专 业 名 称	计算机基础成绩		
				上 机	笔 试	总 分
G01201	程建国	男	计算机应用 2001 级	55	38	93
G01205	吴平	男	计算机应用 2001 级	54	35	89
G00502	朱军	男	商务管理 2000 级	50	30	80
G00100	邵晓云	女	商务英语 2000 级	50	40	90
G01306	马明明	女	船舶应用 2001 级	40	25	65
G01401	林丽	女	汽车技术 2001 级	40	30	70

表 1.3 2001 级计算机应用专业学生情况登记表

学 号	姓 名	性 别	出 生 年 月	籍 贯	家 庭 住 址
G01201	程建国	男	1981.3	江苏	南京
G01202	王玲玲	女	1981.9	福建	福州
G01203	年四	男	1982.6	河北	石家庄
G01204	刘红	女	1981.4	江西	南昌
G01205	吴平	男	1981.11	湖南	长沙
G01206	孟云云	女	1982.12	广东	北京

下面考虑在这两个表中进行查找的问题。

首先考虑在表 1.2 所示的无序表中进行查找。在这个表中，若要查找某位学生的记录，必须从表的第一个记录开始，逐个将表中的记录与所给的学生记录进行比较。若表中的某个学生记录与所给的学生记录完全相同，则查找成功；若表中没有找到所给的学生记录，则查找失败。这种从头至尾逐个在表中查找记录的方法称为顺序查找。显然，在顺序查找中，如果被查找的记录在表的前部，则需要比较的次数就少；如果被查找的记录在表的尾部，则需要比较的次数就多。特别是当要查找的学生记录刚好是登记表中的第一个元素时，只需比较一次就查找成功；但是，当要查找的学生记录刚好是表中最后一个元素时，则需要与表中所有的元素进行比较。当表很大时，顺序查找方法是很费时间的。

现在考虑在表 1.3 所示的有序表中进行查找。由于有序表中的学生记录是按学号从小到大顺序排列的，所以采用有序表的二分查找方法，可以提高查找的效率。

有序表的二分查找方法是：将被查找数与表的中间元素进行比较；若相等，则表示查找成功，结束查找；若被查找数大于表的中间元素，则表示被查找数在表的后半部，此时可以

抛弃表的前半部而保留后半部；若被查找数小于表的中间元素，则表示被查找数在表的前半部，此时可以抛弃表的後半部而保留表的前半部。然后对剩下的部分再按上述方法进行查找。这个过程一直做到在某一次的比较中相等（查找成功）或剩下部分已空（查找失败）为止。这种查找方法称为有序表的二分查找。

需要指出的是：二分查找方法只适用于有序表，无序表是无法进行二分查找的。

在日常生活和学习中也经常用到二分查找。例如，在英文字典中查找某个单词。英文字典按 26 个英文字母顺序排列，是一个有序表。若要查找某个单词，可以根据所查单词的第一个字母，直接翻到大概的位置，然后进行比较，根据比较结果再决定向前或向后翻。重复上述过程，直到找到该单词为止。如果英文字典中的单词不是按 26 个英文字母顺序排列的，而是将几万个单词任意排列，那么为查找某个单词就必须从字典的第一页开始逐页进行查找。

由此可见，数据的组织方式和数据在表中的排列顺序，都会影响查找的效率。

在计算机上实现学生成绩系统的管理，必然要涉及以下三个问题：如何组织学生成绩表？采用何种存储方式将表中的数据及数据之间的关系存放到计算机的存储器中？在计算机上如何完成学生成绩系统的管理功能？例如，查看全部或某位学生的成绩；当学生退学时，删除相应的学生记录；当进来新学生时，增加相应的学生记录；当学生补考时，能够修改相应的成绩记录等。这些就是数据结构所要讨论的问题。

综上所述，我们可以给数据结构下个定义：数据结构就是选择适当的组织方式按照某种关系来组织大量的数据，以一定的存储方式把它们存储到计算机中，并在这些数据上定义一个相应的运算，以提高计算机的数据处理能力的一门学科。

1.2 数据结构的基本概念

在这一节中，我们将对书中一些常用的名词和术语给出确切的定义，以便在今后的学习中能有一个统一的概念。

1.2.1 数据结构的基本术语

(1) **数据 (Data)**：数据是信息的载体，是描述客观事物的数、字符，以及所有能够输入到计算机中并被计算机程序识别和处理的一切对象。例如，解代数方程的程序中所用到的整数和实数，文本编辑中所用到的函数和字符串等，都是计算机程序加工和处理的对象。数据大致可分为两类：一类是数值型数据，包括整数、浮点数、复数、双精度数等，主要用于工程和科学计算，以及商业事务处理；另一类是非数值型数据，主要包括字符和字符串，以及文字、声音、图形、图像等数据。

(2) **数据元素 (Data Element)**：是数据的基本单位，亦称为结点、元素、顶点和记录等，在计算机程序中通常作为一个整体进行考虑和处理。有时一个数据元素可由若干个数据项组成。

数据项 (Data Item)是具有独立意义的最小的数据单位，是对数据元素属性的描述。

(3) **数据对象 (Data Object)**：是具有相同性质的数据元素的集合，是数据的一个子集。例如，整数的数据对象可以是集合 $N=\{0, \pm 1, \pm 2, \dots\}$ ，英文字母组成的数据对象可以是集合 $C=\{'A', 'B', \dots, 'Z'\}$ ，一年四季的名称所组成的数据对象可以是集合 $S=\{\text{春, 夏, 秋, 冬}\}$ ，家庭成员名称所组成的数据对象可以是集合 $F=\{\text{祖父, 父亲, 叔叔, 儿子, 女儿, 孙子, } \dots\}$ 。

在学校中，学生是更复杂的数据对象，它的每一个数据元素就是一个学生记录，每个学生记录包括：学号、姓名、性别、出生年月、家庭住址等数据项，以表明学生在某一方面的属性，参见表 1.3。

在学生选课系统中，可能以一个班级的学生记录作为学生数据对象，也可能以一个年级或一个学校的学生记录作为学生数据对象，参见表 1.2。因此，如何选择数据对象将依据要求不同而定。

(4) **数据类型 (Data Type)**：是具有相同性质的计算机数据的集合和定义在这个数据集合上的一组操作的总称。例如，C 语言中的整数类型，它的值是 $[-\text{MAXINT}, \text{MAXINT}]$ 区间上的整数（区间大小依赖于不同的机器），定义在这个整数集上的操作为：加、减、乘、除和取模等算术运算。

数据类型可以分为两类：**原子数据类型**和**结构数据类型**。原子数据类型是由计算机语言提供的，它的值是不可分解的。例如，C 语言中的基本类型（整型、实型、字符型和枚举类型）、指针类型和空类型。结构数据类型通常是由用户自己定义的，它的值是由若干成分按某种结构组成的，因此是可以分解的，并且它的成分可以是非结构类型的，也可以是结构类型的。例如，C 语言中的数组和结构类型等。

(5) **数据结构 (Data Structure)**：是指某一数据对象及该对象中所有数据元素之间的关系组成。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，一般这三方面为：数据的逻辑结构、数据的存储结构和数据的运算。

一个数据对象中所有数据成员之间一定存在某种联系。数据之间的相互关系，被称为**数据的逻辑结构 (Logical Structure)**。在计算机中存储数据时，不仅要存储数据本身的信息，还要存储各数据之间的前后关系的信息（即逻辑结构）。数据及其关系在计算机中的存储方式，称为**数据的存储结构 (Storage Structure)**，或**数据的物理结构**。

数据的逻辑结构是从解决问题的需要出发，为实现必要的功能所建立的数据结构。它属于用户的视图，是面向问题的，例如，在选修课成绩系统中建立按成绩排列的有序表。数据的逻辑结构是独立于计算机的，它与数据在计算机中的存储位置无关。数据的物理结构是指数据在计算机中如何存放，是数据逻辑结构的物理存储方式，是属于具体实现的视图，是面向计算机的。数据的逻辑结构根据问题所要实现的功能建立，数据的物理结构根据问题所要求的响应速度、处理时间、修改时间、存储空间和单位时间的处理量等建立，是逻辑数据的存储映像。

数据的逻辑结构也可以看成从具体问题抽象出来的数学模型，可用二元组表示：

$$B = (D, R)$$

式中， B 表示数据结构； D 是某个数据对象，是数据元素的集合； R 是 D 中各数据元素之间关系的有限集合，一般用二元组表示。

(6) **数据处理 (Data Processing)**：是指对数据进行检索、插入、删除、合并、排序、统计、简单计算、转换、输入和输出等操作过程。早期的计算机主要用于科学和工程计算，进入 20 世纪 80 年代以后，计算机主要用于数据处理。像计算机情报检索系统、经济信息管理系统、图书管理系统、物资调配系统、银行核算系统、财务管理系统等都是计算机在数据处理领域的具体应用。数据结构是进行数据处理的软件基础。

(7) **数据结构的图形表示**：一个数据结构除了可用二元组表示外，还可以用图形来形象地表示。在数据结构的图形中，每个结点（或顶点）表示一个数据元素，用中间标有元素值的方框表示，两个结点之间用带箭头的连线（称为**有向边**或**弧**）表示对应关系中的一个序偶，

其中序偶的第一个元素（称为第二个元素的前驱结点）为有向边的起始结点，第二个元素（称为第一个元素的后继结点）为有向边的终止结点，即箭头所指的结点。

1.2.2 数据的逻辑结构

数据元素相互之间的关系，称为结构（Structure）。根据数据元素之间不同的关系特性，数据元素间的关系分为 4 种类型：集合、线性结构、树状结构及图状或网状结构。通常数据的逻辑结构分为两大类：线性结构和非线性结构。

1. 线性结构

线性结构的逻辑特征是：线性结构中每一个数据元素（除第一个和最后一个数据元素外），仅有一个直接前驱和一个直接后继；而第一个数据元素只有一个直接后继，没有直接前驱；最后一个数据元素只有一个直接前驱，没有直接后继。

线性结构也称为线性表。选修课成绩表（见表 1.2）和学生情况登记表（见表 1.3）就是典型的线性结构。本书将在第 2~4 章中介绍数据的线性结构。

2. 非线性结构

非线性结构的逻辑特征是：一个数据元素可能有多个直接前驱和直接后继。树状结构和图结构都是典型的非线性结构。本书将在第 6 章和第 7 章中介绍树状结构和图结构。

3. 数据的逻辑结构实例

为了增加对数据结构的感性认识，下面举例说明数据结构这一重要概念。

【例 1.3】一年四季名称所组成的数据结构可以表示为：

$$B=(D, R)$$

$$D=\{\text{春}, \text{夏}, \text{秋}, \text{冬}\}$$

$$R=\{\langle \text{春}, \text{夏} \rangle, \langle \text{夏}, \text{秋} \rangle, \langle \text{秋}, \text{冬} \rangle\}$$

该结构的数据元素是一年中四个季节的名称。各季节间的顺序关系是：“春”是“夏”的直接前驱，“夏”是“秋”的直接前驱，而“秋”是“冬”的直接前驱，反之“夏”是“春”的直接后继。在该结构中，数据元素之间是一一对应的关系，即线性关系。我们把具有这种特点的数据结构称为线性结构。

图 1.3 是一年四季名称数据结构的图形表示。在数据结构的图形表示中，数据集合 D 中的每个数据元素用中间标有元素值的方框表示，数据元素间的关系用带箭头的线段表示。

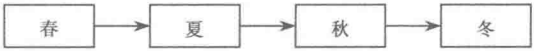


图 1.3 线性数据结构示例：一年四季名称数据结构的图形表示

【例 1.4】考试成绩统计表见表 1.4。

表 1.4 中的每一行是一名考生的记录，每个记录由准考证号、姓名、各科成绩及总分等数据项组成。其中，准考证号为该记录的主关键字。若将所有考生记录按总分从高到低排列，则高考成绩统计表是一个按成绩排列的有序表。表中每个学生记录所组成的数据结构可表示为：

$$B=(D, R)$$

$$D=\{\text{胡桃}, \text{黎明}, \text{肖红}, \text{唐平}, \text{程程}, \text{房芳}\}$$

$R=\{ \langle \text{胡桃}, \text{黎明} \rangle, \langle \text{黎明}, \text{肖红} \rangle, \langle \text{肖红}, \text{唐平} \rangle, \langle \text{唐平}, \text{程程} \rangle, \langle \text{程程}, \text{房芳} \rangle \}$

表 1.4 考生成绩统计表

准考证号	姓 名	语 文	外 语	数 学	物 理	化 学	总 分
20006	胡桃	110	135	140	130	120	635
20004	黎明	100	130	140	116	122	608
20001	肖红	120	100	126	100	120	566
20003	唐平	90	130	130	100	95	545
20002	程程	100	110	120	90	100	520
20005	房芳	80	100	90	80	85	435

这是一个典型的线性结构。考生成绩统计表是一个线性表，线性表中所有数据元素按成绩从高到低顺序排列。线性表中第一个记录没有直接前驱，称为开始结点，而最后一个记录没有直接后继，称为终端结点。除第一个记录和最后一个记录外，其他记录仅有一个直接前驱结点和一个直接后继结点。例如，“胡桃”没有前驱结点故为开始结点，而“房芳”没有后继结点是终端结点；“黎明”的前驱结点是“胡桃”，其后继结点是“肖红”。

图 1.4 就是考生成绩统计表这个数据结构对应的图形表示。

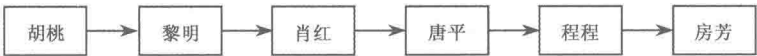


图 1.4 线性数据结构示例：考生成绩统计表的图形表示

【例 1.5】假设家庭成员组成的数据结构可以表示成：

$B=(D,R)$

$D=\{\text{祖父}, \text{叔叔}, \text{父亲}, \text{儿子}, \text{女儿}, \text{孙子}\}$

$R=\{ \langle \text{祖父}, \text{父亲} \rangle, \langle \text{祖父}, \text{叔叔} \rangle, \langle \text{父亲}, \text{儿子} \rangle, \langle \text{父亲}, \text{女儿} \rangle, \langle \text{儿子}, \text{孙子} \rangle \}$

这是一个典型的树状数据结构。组成这个结构的数据元素是家庭成员名，在考虑家庭成员间的辈分关系时，则“祖父”是“父亲”的直接前驱，而“儿子”和“女儿”都是“父亲”的直接后继；“儿子”是“孙子”的直接前驱，而“孙子”则是“儿子”的直接后继。

图 1.5 是家庭成员间辈分关系数据结构的图形表示。该图形就像一棵倒画的树。在这棵树中，最上面一个没有前驱的结点称为根结点，最下面一层只有前驱没有后继的结点称为叶结点。在一棵树中，每个结点有且仅有一个直接前驱结点（除第一个结点称为树的根结点外），但可以有任意多个后继结点。这种数据结构的特点是，数据元素之间是一对多的关系，即层次关系。我们把具有这种特点的数据结构称为树状结构，简称为树。

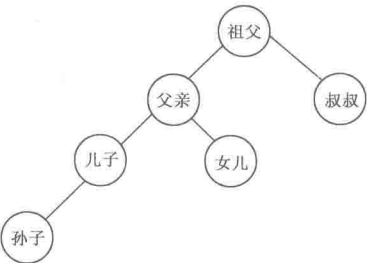


图 1.5 树状数据结构示例：家庭成员间辈分关系的图形表示

【例 1.6】假设国内若干城市之间的航线组成的数据结构可表示成：

$B=(D, R)$

$D=\{\text{北京, 上海, 武汉, 香港, 重庆, 广州}\}$

$R=\{r_1, r_2, r_3, r_4\}$

$r_1=\{(\text{北京, 上海}), (\text{北京, 香港}), (\text{北京, 广州}), (\text{北京, 重庆}), (\text{北京, 武汉})\}$

$r_2=\{(\text{上海, 香港}), (\text{上海, 重庆}), (\text{上海, 武汉}), (\text{上海, 广州})\}$

$r_3=\{(\text{武汉, 香港}), (\text{武汉, 重庆}), (\text{武汉, 广州})\}$

$r_4=\{(\text{香港, 重庆}), (\text{香港, 广州}), (\text{广州, 重庆})\}$

图 1.6 是国内若干城市间部分航线示意图，这是一个典的图数据结构。组成这种数据结构的数据元素是城市名称（顶点），在图中，每个城市（顶点）都可以与若干城市（顶点）相连，是一种多对多的关系。在航线图中，对每个城市的描述可以用一个顶点来表示，而每个城市的基本信息，如城市的名称、机场的位置、航线的多少等则可以用顶点中的数据项来描述。这些描述在航线图中被省略了。

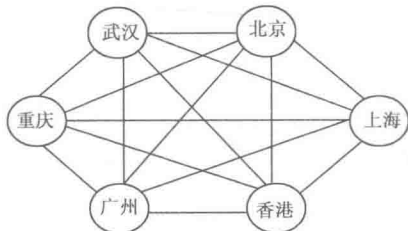


图 1.6 图数据结构示例：国内若干城市间部分航线示意图

从图 1.6 中可以看出， R 是 D 上的对称关系，这种结构的特点是数据元素之间的联系是多对多的关系，即**网状关系**；也就是说，每个结点可以有任意多个前驱结点和任意多个后继结点。我们把具有这种特点的数据结构称为**图结构**，简称**图**。

1.2.3 数据的存储结构

数据处理是计算机应用的一个重要领域。在实际进行数据处理时，所有要处理的数据都要存放到计算机的存储器中。数据在计算机中的存储方式有多种：顺序、链接、索引和散列等。因此，同一种数据的逻辑结构可以根据需要表示成任意一种或几种不同的存储结构。下面将简要介绍常用的 4 种存储方法。

1. 顺序存储方法

顺序存储方法是将逻辑上相邻的结点存储在物理位置上亦相邻的存储单元里，也就是将所有存储结点相继存放在一个连续相邻的存储区里。用存储结点间的位置关系来表示结点之间的逻辑关系。因此，顺序存储结构只需要存储结点的信息，不需要存储结点之间的关系。

计算机的存储器是由很多存储单元（Word 或 Byte）组成的，每个存储单元都有唯一的地址（编号）。每个存储单元的地址编号都是线性连续的。我们把两个互为前驱、后继的存储单元称为相邻存储单元，把一片相邻的存储单元称为存储区域。

顺序存储结构是一种最基本的存储方法，通常可用程序设计语言中的数组来实现。

【例 1.7】请用顺序存储方式表示一周 7 天，假设一周 7 天的数据结构为：

$B=(D, R)$

$D=\{Sun, Mon, Tue, Wed, Thu, Fri, Sat\}$

$R=\{<Sun, Mon>, <Mon, Tue>, <Tue, Wed>, <Wed, Thu>, <Thu, Fri>, <Fri, Sat>\}$

若采用顺序存储方法将一周 7 天存储在计算机中，其顺序存储结构如图 1.7 所示。

存储地址	数组的下标	数据域
1000	0	Sun
1001	1	Mon
1002	2	Tue
1003	3	Wed
1004	4	Thu
1005	5	Fri
1006	6	Sat

图 1.7 线性结构的顺序存储结构示例

2. 链接存储方法

链接存储方法是在存储每个结点信息的同时，需要增加一个指针来表示结点间的逻辑关系。该方法不要求逻辑上相邻的结点在物理位置上亦相邻，结点间的逻辑关系是由附加的指针字段表示的。因此，链接存储结构中的每个结点由两部分组成：一部分用于存储结点本身的信息，称为**数据域**；另一部分用于存储该结点的后继结点（或前驱结点）的存储单元地址，称为**指针域**。指针域可以包含一个或多个指针，这由结点之间的关系所决定。

链接存储结构通常借助于程序设计语言中的指针来实现。

【例 1.8】请将一周 7 天这个数据结构采用链接存储方式来表示。

该数据结构的链接存储结构如图 1.8（a）所示，图 1.8（b）是该链接存储结构的图形表示。在图中，方框表示一个结点，框中数字为该结点的值，箭头代表指针，它表示各结点之间的关系。



(a) 链接存储结构示意图



(b) 链接存储结构的图形表示

图 1.8 线性结构的链接存储结构示例

3. 索引存储方法

索引存储方法是在存储结点信息的同时，建立一个附加的索引表。索引表中每一项称为一个索引项。索引项的一般形式是：（关键字，地址），关键字是能唯一标识一个结点的数据