



21世纪高职高专规划教材

计算机系列

数据结构概论

尹绍宏 董卿霞 苑春苗 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社

<http://press.bjtu.edu.cn>

21 世纪高职高专规划教材·计算机系列

数据结构概论

尹绍宏 董卿霞 苑春苗 编著

清华大学出版社

北京交通大学出版社

·北京·

内 容 简 介

本书详细地介绍了各种类型的数据结构，以及查找和排序的方法。对每一种数据结构，主要讲述其基本概念，各种存储结构，以及不同存储结构下的各种操作的实现，并用 C 语言对其算法进行实现。对查找和排序的各种不同方法除讲述其方法外，还给出了用 C 语言实现的算法程序，并对不同的算法给出了定性的分析和比较。

本书既注重理论又注重实践，并配有大量的习题和实习题，内容丰富、概念清楚、通俗易懂，既可用于教学，又便于读者自学。

本书可以作为大专院校计算机应用及相关专业的教材，也可以供从事计算机技术与应用工作的科技人员使用。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目 (CIP) 数据

数据结构概论 / 尹绍宏, 董卿霞, 苑春苗编著. —北京: 清华大学出版社; 北京交通大学出版社, 2004.5

(21 世纪高职高专规划教材·计算机系列)

ISBN 7-81082-289-6

I. 数… II. ①尹…②董…③苑… III. 数据结构—高等学校: 技术学校—教材
IV. TP311.12

中国版本图书馆 CIP 数据核字 (2004) 第 018557 号

责任编辑: 韩 乐 特邀编辑: 朱 宇

出版者: 清华大学出版社 邮编: 100084 电话: 010-62776969

北京交通大学出版社 邮编: 100044 电话: 010-51686045, 62237564

印刷者: 北京人卫印刷厂

发行者: 新华书店总店北京发行所

开 本: 185×260 印张: 15 字数: 374 千字

版 次: 2004 年 5 月第 1 版 2004 年 5 月第 1 次印刷

书 号: ISBN 7-81082-289-6 / TP · 110

印 数: 0001~5000 册 定价: 21.00 元

21 世纪高职高专规划教材·计算机系列 编审委员会成员名单

主任委员 李兰友 边奠英

副主任委员 周学毛 崔世钢 王学彬 丁桂芝 赵伟
韩瑞功 汪志达

委 员 (按姓名笔画排序)

马 辉	万志平	万振凯	王永平	王建明
尤晓晔	丰继林	尹绍宏	左文忠	叶 华
叶 伟	付晓光	付慧生	冯平安	江 中
佟立本	刘 炜	刘建民	刘 晶	曲建民
孙培民	邢素萍	华铨平	吕新平	陈小东
陈月波	李长明	李 可	李志奎	李 琳
李源生	李群明	李静东	邱希春	沈才梁
宋维堂	汪 繁	张文明	张权范	张宝忠
张家超	张 琦	金忠伟	林长春	林文信
罗春红	苗长云	竺士蒙	周智仁	孟德欣
柏万里	宫国顺	柳 炜	钮 静	胡敬佩
姚 策	赵英杰	高福成	贾建军	徐建俊
殷兆麟	唐 健	黄 斌	章春军	曹豫莪
程 琪	韩广峰	韩其睿	韩 劼	裘旭光
童爱红	谢 婷	曾瑶辉	管致锦	熊锡义
潘玫玫	薛永三	操静涛	鞠洪尧	

出版说明

高职高专教育是我国高等教育的重要组成部分，它的根本任务是培养生产、建设、管理和服务第一线需要的德、智、体、美全面发展的高等技术应用型专门人才，所培养的学生在掌握必要的基础理论和专业知识的基础上，应重点掌握从事本专业领域实际工作的基本知识和职业技能，因而与其对应的教材也必须有自己的体系和特色。

为了适应我国高职高专教育发展及其对教学改革和教材建设的需要，在教育部的指导下，我们在全中国范围内组织并成立了“21世纪高职高专教育教材研究与编审委员会”（以下简称“教材研究与编审委员会”）。“教材研究与编审委员会”的成员单位皆为教学改革成效较大、办学特色鲜明、办学实力强的高等专科学校、高等职业学校、成人高等学校及高等院校主办的二级职业技术学院，其中一些学校是国家重点建设的示范性职业技术学院。

为了保证规划教材的出版质量，“教材研究与编审委员会”在全国范围内选聘“21世纪高职高专规划教材编审委员会”（以下简称“教材编审委员会”）成员和征集教材，并要求“教材编审委员会”成员和规划教材的编著者必须是从事高职高专教学第一线的优秀教师或生产第一线的专家。“教材编审委员会”组织各专业的专家、教授对所征集的教材进行评选，对列选教材进行审定。

目前，“教材研究与编审委员会”计划用2~3年的时间出版各类高职高专教材200种，范围覆盖计算机应用、电子电气、财会与管理、商务英语等专业的主要课程。此次规划教材全部按教育部制定的“高职高专教育基础课程教学基本要求”编写，其中部分教材是教育部《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》的研究成果。此次规划教材编写按照突出应用性、实践性和针对性的原则编写并重组系列课程教材结构，力求反映高职高专课程和教学内容体系改革方向；反映当前教学的新内容，突出基础理论知识的应用和实践技能的培养；适应“实践的要求和岗位的需要”，不依照“学科”体系，即贴近岗位群，淡化学科；在兼顾理论和实践内容的同时，避免“全”而“深”的面面俱到，基础理论以应用为目的，以必要、够用为度；尽量体现新知识、新技术、新工艺、新方法，以利于学生综合素质的形成和科学思维方式与创新能力的培养。

此外，为了使规划教材更具广泛性、科学性、先进性和代表性，我们希望全国从事高职高专教育的院校能够积极加入到“教材研究与编审委员会”中来，推荐“教材编审委员会”成员和有特色、有创新的教材。同时，希望将教学实践中的意见与建议及时反馈给我们，以便对已出版的教材不断修订、完善，不断提高教材质量，完善教材体系，为社会奉献更多更新的与高职高专教育配套的高质量教材。

此次所有规划教材由全国重点大学出版社——清华大学出版社与北京交通大学出版社联合出版。适合于各类高等专科学校、高等职业学校、成人高等学校及高等院校主办的二级职业技术学院使用。

21世纪高职高专教育教材研究与编审委员会
2004年3月

前 言

随着计算机技术的不断发展,计算机在各个领域都得到了广泛的应用。但在应用过程中,都会涉及数据的组织与程序的编写等问题,都会用到各种各样的数据结构,特别是对非数值型数据的表示,各种操作算法的实现,都离不开数据结构课程的内容。因此,数据结构一直是各高等院校计算机专业教学内容中的一门主要的专业基础课程。

本书是在作者多年教学经验的基础上编写而成的。在编写过程中,从内容和结构入手,进行了精心的设计。在内容选择方面,全面地反映了数据结构各个方面的内容,突出了概念、方法和应用,既注重基本原理的介绍,又注重实践能力的培养;在结构的安排方面,以逻辑结构为主要线索,对每一种数据结构都是按照基本概念和定义,顺序存储结构的表示及各基本操作的实现和链式存储结构的表示及各基本操作的实现,以及应用这几个方面详细地进行介绍。这样便于对各种结构的学习,掌握各种结构的特点及它们之间的关系。书中还配有大量的例题、习题和实习题供学生练习,以加深对各知识点的理解。书中的算法全部采用C语言描述,所给出的程序均已在计算机上运行通过,并给出了程序的运行结果,以便读者了解算法的实质和基本思想。

全书分为10章,第1章介绍了数据、数据结构、抽象数据类型及算法的概念和性能分析等基本概念;第2章至第6章主要讨论了各种线性结构及其应用,包括线性表、栈、队列、串、二维数组和广义表;第7章和第8章讨论了非线性结构及其应用,重点讨论了树和二叉树、图的基本概念及其应用;第9章讨论了各种内部排序的方法,以及每种排序方法的性能、特点和适用场合;第10章讨论了不同存储结构下的各种查找方法及不同查找方法的性能分析。在每一章的后面都附有习题和实习题,可以检验读者的学习效果,培养程序设计能力。

王炜、杨清永、高海明、王宇鑫、苏丽华、赵金英、黎剑兵等也参加了程序的调试及部分文档的整理工作。

由于作者水平有限,书中难免会有错漏和疏忽,殷切希望广大读者予以批评指正。

编者

2004年4月

目 录

第 1 章 绪论	1
1.1 基本概念和术语	1
1.2 发展历程	4
1.3 算法和算法描述	5
1.3.1 概念和特性	5
1.3.2 算法设计要求	5
1.3.3 算法描述	6
1.4 算法的性能分析	7
1.4.1 时间复杂度	7
1.4.2 空间复杂度	9
小结	9
习题	9
实习	10
第 2 章 线性表	11
2.1 概念和定义	11
2.1.1 概念	11
2.1.2 定义	12
2.2 顺序存储结构	12
2.2.1 顺序表的存储表示	13
2.2.2 顺序表的基本操作的实现	13
2.3 链式存储结构	17
2.3.1 单链表的存储表示	17
2.3.2 单链表基本操作的实现	18
2.3.3 循环链表的表示和基本操作的实现	22
2.3.4 双向链表的表示和基本操作的实现	22
2.4 应用举例	24
2.4.1 顺序表	24
2.4.2 单链表	26
小结	28
习题	29
实习	31
第 3 章 栈	32
3.1 概念和定义	32

3.2	顺序存储表示	33
3.2.1	顺序栈的存储表示	33
3.2.2	顺序栈基本操作的实现	34
3.3	链式存储结构	36
3.3.1	链栈的存储表示	36
3.3.2	链栈基本操作的实现	36
3.4	应用举例	38
	小结	42
	习题	42
	实习	44
第4章	队列	45
4.1	概念和定义	45
4.2	顺序存储结构	46
4.2.1	顺序队列的存储表示	46
4.2.2	顺序队列基本操作的实现	46
4.2.3	循环队列	49
4.3	链式存储结构	50
4.3.1	链队列的存储表示	50
4.3.2	链队列基本操作的实现	51
4.4	应用举例	53
	小结	54
	习题	55
	实习	56
第5章	串	57
5.1	概念和定义	57
5.2	顺序存储结构	59
5.2.1	定长顺序串的存储表示及操作的实现	59
5.2.2	堆存储表示及操作的实现	61
5.3	块链存储表示	65
5.4	应用举例	65
	小结	67
	习题	67
	实习	68
第6章	二维数组和广义表	69
6.1	二维数组概念和定义	69
6.2	二维数组的顺序存储结构	70
6.3	矩阵的压缩存储	70
6.3.1	概念	71
6.3.2	特殊矩阵的压缩存储	71

6.3.3	稀疏矩阵的顺序存储表示和基本操作的实现	72
6.3.4	稀疏矩阵的链式存储表示和基本操作的实现	79
6.4	广义表的概念和定义	82
6.5	广义表的操作和链式存储结构	82
	小结	85
	习题	86
	实习	88
第7章	树与二叉树	89
7.1	树的概念	89
7.1.1	定义	89
7.1.2	表示方法	90
7.1.3	基本概念和常用术语	91
7.2	二叉树	92
7.2.1	概念和定义	92
7.2.2	性质	93
7.2.3	存储结构	94
7.2.4	遍历	95
7.2.5	二叉树的线索化	99
7.3	树和森林	104
7.3.1	树的存储结构	104
7.3.2	树和森林的遍历	106
7.3.3	树、森林与二叉树的转换	107
7.4	哈夫曼树	108
7.4.1	概念和定义	108
7.4.2	哈夫曼树的构造	109
7.4.3	哈夫曼编码的实现	112
	小结	115
	习题	115
	实习	116
第8章	图	118
8.1	图的概念	118
8.1.1	定义	118
8.1.2	基本概念和常用术语	119
8.2	存储结构	120
8.2.1	邻接矩阵表示及各操作的实现	121
8.2.2	邻接表的表示及各操作的实现	125
8.3	图的遍历	131
8.3.1	深度优先搜索	131
8.3.2	广度优先搜索	134

8.4	生成树和最小生成树	135
8.4.1	生成树的概念和分类	135
8.4.2	最小生成树的概念和实现方法	136
8.5	AOV 网及其应用	143
8.5.1	概念	143
8.5.2	拓扑排序	143
8.6	AOE 网及其应用	148
8.6.1	概念	148
8.6.2	关键路径	148
8.7	最短路径	155
8.7.1	任意源点到其余各点的最短路径	155
8.7.2	任意两点间的最短路径	159
	小结	160
	习题	160
	实习	162
第 9 章	排序	163
9.1	概念及分类	163
9.2	插入排序	164
9.2.1	直接插入排序	164
9.2.2	折半插入排序	167
9.2.3	2-路插入排序	169
9.2.4	希尔排序	170
9.3	交换排序	172
9.3.1	冒泡排序	173
9.3.2	快速排序	175
9.4	选择排序	178
9.4.1	简单选择排序	178
9.4.2	树型选择排序	181
9.4.3	堆排序	182
9.5	K-路归并排序	186
9.6	基数排序	190
9.7	内部排序方法的比较	191
9.7.1	时间性能	192
9.7.2	空间性能	192
9.7.3	稳定性	192
9.7.4	排序方法的选择	193
	小结	193
	习题	194
	实习	195

第 10 章 查找	196
10.1 概念	196
10.2 顺序存储结构查找	198
10.2.1 顺序查找	198
10.2.2 折半查找	200
10.2.3 分块查找	203
10.3 树存储结构查找	206
10.3.1 二叉排序树	206
10.3.2 B-树	210
10.4 哈希表查找	212
10.4.1 基本概念	213
10.4.2 哈希函数的构造方法	213
10.4.3 解决冲突的方法	215
10.4.4 查找方法	218
小结	221
习题	221
实习	222
习题答案	223
参考文献	225

第1章 绪 论

本章要点:

- 数据结构的基本概念和术语
 - 数据结构的发展过程及其所处地位
 - 算法和算法的描述
 - 算法的性能分析
-

1.1 基本概念和术语

数据结构是计算机专业的专业基础课之一，是一门十分重要的核心课程。计算机的所有系统软件和应用软件都需要用到各种类型的数据结构。要想编写出一个好的程序，仅仅学习计算机语言是不够的，必须扎实地掌握数据结构的基本知识和基本技能。同时，它也是学习计算机专业其他课程，如操作系统、数据库原理与应用和软件工程等所必需的基础知识。

掌握好数据结构方面的知识，对于增强我们解决实际问题的能力将会有很大帮助。事实上，一个好的程序无非是选择了一个合理的数据结构和一个好的算法，而好的算法的选择在很大程度上取决于描述实际问题的数据结构。所以，学好数据结构课程是进一步提高程序设计水平的关键之一。

随着计算机应用领域的不断扩大，非数值计算问题已跃居主导地位，简单的数据类型已远远不能满足需要，各数据元素之间的复杂联系不再是普通数学方程式所能表达的了。因此简单地说，数据结构就是研究非数值计算的程序设计问题中，计算机的操作对象，以及它们之间的关系和操作等的学科。下面将对一些在以后章节中使用的基本概念和术语加以定义和解释。

1. 数据

数据是人们利用文字符号、数字符号及其他规定的符号，对现实世界的事物及其活动所做的抽象描述。例如，文字、数字和符号都是数据。

2. 数据元素

数据元素是数据的基本单位。例如，学生是一个数据元素。有些时候，一个数据元素可以由若干个数据项组成，数据项是具有独立含义的数据最小单位，称为字段或域。例如，作为数据元素的学生由学号、姓名、性别和班级等数据项组成。

3. 数据对象

数据对象是性质相同的数据元素的集合，是数据的一个子集。例如，整数数据对象是集

合 $\{0, \pm 1, \pm 2, \dots\}$, 字符数据对象是集合 $\{ 'A', 'B', \dots, 'Z' \}$ 。

4. 数据结构

数据结构是指数据及其相互之间的联系, 可以看做是相互间存在一种或多种特定关系的数据元素的集合。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关系, 这种数据元素之间的关系称为结构。根据数据元素间的不同特性关系, 通常有下面 4 类基本结构。

(1) 集合结构

集合结构的数据元素之间的关系是“属于同一个集合”, 它是元素间关系最为松散的一种结构。如图 1-1(a)所示。

(2) 线性结构

线性结构的数据元素之间存在一对一的关系, 如图 1-1(b)所示。

(3) 树型结构

树型结构的数据元素之间存在一对多的关系, 如图 1-1(c)所示。

(4) 图状结构

图状结构的数据元素之间存在多对多的关系, 图状结构又称网状结构, 如图 1-1(d)所示。

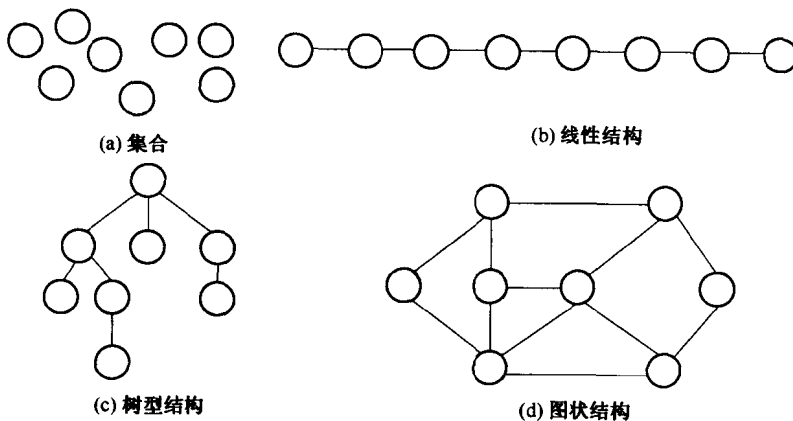


图 1-1 基本数据结构图

5. 数据类型

数据类型是与数据结构密切相关的一个概念, 容易引起混淆。数据类型最早出现在高级语言中, 是对数据的取值范围, 每一数据的结构, 以及允许执行的操作的一种描述, 它刻画了程序中操作对象的特性。每一种程序语言都定义有自己的数据类型, 在用程序语言编写的程序中, 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。所谓数据类型, 是一个值的集合, 以及在这些值上定义的一组操作的总称。

数据类型可以分为简单类型和结构类型两种。简单类型中的每个数据(即简单数据)都是无法再分割的整体, 如一个整数、实数、字符、指针、枚举量等都是无法再分割的整体, 所以它们所属的类型均为简单类型。结构类型由简单类型按照一定的规则构造而成, 并且结构类型中可以包含结构类型, 所以一种结构类型中的数据(即结构数据)可以分解为若干个简单数据或结构数据, 每个结构数据仍可再分。例如, C 语言中的数组是一种结构类型, 它

是由固定个数的同一类型的数据顺序排列而成。结构体也是一种结构类型，它是由固定个数的不同类型的数据顺序排列而成。

数据类型也可以被定义为一种数据结构和能够对该数据结构进行的操作的集合。对于简单类型，其数据结构就是相应取值范围内的所有数据，每一个数据值是不可分的独立整体，因而数据值内部也就无结构可言。对于结构类型，其数据结构就是相应元素的集合。

6. 抽象数据类型

抽象数据类型 (Abstract Data Type, ADT) 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仅取决于它的一组逻辑特性，而与其在计算机内部的表示方式及实现无关，即它独立于具体实现。一个 ADT 定义可描述为

```

ADT 抽象数据类型名 {
    数据对象: (数据对象的定义)    /* 对数据元素的说明*/
    数据关系: (数据关系的定义)    /* 对数据元素之间逻辑关系的描述*/
    基本操作: (基本操作的定义)    /* 对该抽象数据类型允许进行的操作的描述*/
        操作 1                      /* 对本操作的初始条件、参数的类型、操作的功能
                                   和操作结果的描述*/
        操作 2
        .....
} ADT 抽象数据类型名

```

抽象数据类型可以看做是描述问题的模型，其优点是将数据和操作封装在一起，使用户程序只能通过在 ADT 里定义的某些操作来访问其中的数据，从而实现了信息隐蔽。

数据结构包括数据的逻辑结构和物理结构。数据的逻辑结构可以看做是从具体问题中抽象出来的数学模型，与数据的存储无关，是独立于计算机的。数据元素之间的逻辑关系称为数据的逻辑结构。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称为数据的物理结构，又称为数据的存储结构。它研究的是数据结构在计算机中的实现方法，包括数据元素及其关系在计算机中的表示。

数据的逻辑结构有线性结构和非线性结构两大类。

线性结构的逻辑特征是，若结构是非空集，则有且只有一个开始结点和一个终端结点，并且除第一个结点之外，每个结点都有且只有一个直接前驱，除最后一个结点之外，每个结点都有且只有一个直接后继。线性表就是典型的线性结构。

非线性结构的逻辑特征是，一个结点可能有多个直接前驱和多个直接后继。树和图的结构就是典型的非线性结构。

数据的存储结构可以采用顺序存储、链式存储、索引存储和散列存储 4 种方法。

顺序存储方法是把逻辑上相邻的结点存储在物理位置上相邻的存储单元中，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。顺序存储结构一般借助于程序设计语言的数组来描述。

链式存储方法不要求逻辑上相邻的结点在物理位置上亦相邻，其结点间的逻辑关系是由附加的指针域表示的。由此得到的存储表示称为链式存储结构。链式存储结构一般借助于程序设计语言的指针类型来描述。

索引存储方法通常是在存储结点信息的同时建立一个索引表，索引表中的每一项称为索引项，它一般包括关键字和地址两项内容，关键字是唯一能标识一个结点的数据项。建立索引表时，可以是每个结点在索引表中都有一个索引项，也可以是一组结点在索引表中只对应于一个索引项。索引存储方法便于检索操作。

散列存储方法的基本思想是根据结点的关键字直接计算出该结点的存储地址。

上述 4 种基本的存储方法，既可以单独使用，也可以混合起来对数据进行存储使用。对于同一种逻辑结构，采用不同的存储方法，可以得到不同的存储结构，应根据具体要求而定。

1.2 发展历程

数据结构作为一门独立的课程是从 1968 年才开始设立的。在此之前，它的某些内容曾在其他课程中有所阐述。1968 年，在美国一些大学的计算机系的教学计划中，虽然把数据结构规定为一门课程，但对课程的范围仍没有明确的规定。当时，数据结构几乎与图论，特别是与表、树的理论为同义语。随后，数据结构的概念被扩充到包括网络、集合代数论、关系等方面，从而变成了现在称之为离散结构的内容。由于数据必须在计算机中进行处理，因此，不仅需要考虑数据本身的数学性质，而且还必须考虑数据的存储结构，这就进一步扩大了数据结构的内容。近年来，随着数据库系统的不断发展，在数据结构课程中又增加了文件管理（特别是大型文件的组织等）的内容。

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》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，结构化程序设计成为程序设计方法学的主要内容，人们就越来越重视数据结构，认为程序设计的实质是对确定的问题选择一种好的结构，再加上设计一种好的算法。从 70 年代中期到 80 年代初，各种版本的数据结构著作相继出现。

目前在我国，数据结构也已经不仅仅是计算机专业的教学计划中的核心课程之一，而是其他非计算机专业的主要选修课程之一。

数据结构在计算机科学中是一门综合性的专业基础课。数据结构的研究不仅涉及计算机硬件（特别是编码理论、存储装置和存取方法等）的研究范围，而且与计算机软件的研究有着更密切的关系，无论是编译原理还是操作系统，都涉及数据元素在存储器中的分配问题。在研究信息检索时也必须考虑如何组织数据，以便数据元素更为方便地查找和存取。因此，可以认为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核心课程。在计算机科学中，数据结构不仅是一般程序设计（特别是非数值计算的程序设计）的基础，而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的重要基础。

今后，数据结构还将继续发展，一方面，向各专门领域中特殊问题的数据结构的研究方向发展，如多维图形数据结构等；另一方面，从抽象数据类型的观点讨论数据结构，这一发展方向已成为新的趋势，越来越为人们所重视。

1.3 算法和算法描述

计算机软件的最终结果都是以程序形式表现的，数据结构的各种操作都是以算法形式描述的。数据结构、算法和程序是密不可分的，它们之间的关系可以表示为

$$\text{数据结构} + \text{算法} = \text{程序}$$

1.3.1 概念和特性

数据元素之间具有逻辑关系和物理关系。与此对应，有逻辑结构上的操作功能和具体存储结构上的操作实现。我们把具体存储结构上的操作实现方法称为算法。

确切地说，算法是对特定问题求解步骤的一种描述，它是指令的有限序列，其中每一条指令表示计算机的一个或多个操作。一个算法具有以下5个重要特性。

- 有穷性。一个算法必须总是（对任何合法的输入值）在执行有穷步之后结束，且每一步都可以在有穷时间内完成。
- 确定性。算法中的每一条指令都必须具有确切的含义，不会产生二义性。
- 可行性。算法中描述的操作都可以通过已经实现的基本运算执行有限次来实现。
- 输入。一个算法有零个或多个输入。这些输入取自于某个选定的对象的集合。
- 输出。一个算法有一个或多个输出。这些输出是与输入有某些特定关系的量。没有输出的算法是没有意义的。

算法与程序不同，程序可以不满足上述的有穷特性。例如，一个操作系统（如 DOS 或 Windows）在用户未操作之前一直处于“等待”的循环中，直到出现新的用户操作为止。但一般的程序则不能出现这种情况。另外，程序中的指令必须是机器可执行的，而算法中的指令则无此限制。算法代表了对问题的解，程序则是算法在计算机上特定的实现。一个算法若用程序设计语言来描述，它就是一个程序。

在计算机科学的研究中，算法与数据结构是相辅相成的。解决某一特定类型问题的算法可以选定不同的数据结构，而选择是否恰当将直接影响算法的效率。

1.3.2 算法设计要求

算法与数据结构的优劣直接相关。设计一个好的算法通常需要满足以下要求。

1. 正确性

一个正确的算法是指在合法的数据输入下，要求算法能在有限的运行时间内得出正确的结果。这是最重要也是最基本的准则。

2. 可使用性

要求算法能够很方便地使用。该特性称为用户友好性。

3. 可读性

算法最主要的目的是阅读与交流，并将它转换成可实现的程序在计算机中执行。算法应当是可读的，即可读性好。在保证算法正确的前提下，应该强调算法的可读性。为了达到这个要求，算法的逻辑必须是清晰的、简单的和结构化的。

4. 健壮性

要求算法具有很好的容错性，即能够对错误的情况进行处理，对不合理的数据进行检查，并做出适当的反应或进行处理，不会经常出现异常中断或死机现象。

5. 效率

算法的效率主要指算法执行时计算机资源的消耗，包括存储单元和运行时间的开销。前者称为算法的空间复杂度，后者称为算法的时间复杂度。对于同一个问题，在相同的数据规模下，如果有多个算法可以使用，则执行时间短的算法效率最高，所需存储空间少的算法较好。通常，两者是矛盾的一对，节约算法的执行时间，往往以牺牲更多的空间作为代价；而为了节省空间可能需要耗费更多的计算时间，因此我们只能根据具体情况有所侧重。若程序使用较少，则力求算法简单易懂；对于需要反复多次使用的程序，应尽可能选用快速的算法；若所求问题的数据量较大，机器的存储空间又较小，则在设计算法时应主要考虑如何节省空间。

1.3.3 算法描述

算法的描述有多种方式，可以用自然语言、计算机程序语言或其他语言来说明，但要求该说明必须能够精确地描述计算过程。一般地，描述算法最恰当的语言是介于自然语言和程序语言之间的伪语言，它的控制结构比较容易理解和表达。在这里，考虑到易于上机验证算法和提高读者的实际程序设计能力，本书中我们采用 C 语言描述算法。C 语言是一种高效、灵活和精炼的高级程序设计语言，优点是类型丰富、语句简捷，编写的程序结构化程度高、可读性强。

常用的用于描述算法的 C 语言语句如下。

(1) 输入语句

```
scanf(<格式控制字符串>, <输入项表>);
```

(2) 输出语句

```
printf(<格式控制字符串>, <输出项表>);
```

(3) 赋值语句

```
变量名 = 表达式;
```

(4) 条件语句

```
if(<条件>)<语句>;
```

或者

```
if(<条件>)<语句 1> else <语句 2>;
```

(5) 循环语句

```
while(<表达式>)<循环体语句>;
```

或者

```
do{
```