



新世纪高等学校规划教材·计算机专业核心课程系列

数据结构与算法

(C++语言描述)

(第二版)

陈明◎编著

SHUJU JIEGOU
YU SUANFA



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社



新世纪高等学校规划教材·计算机专业核心课程系列

数据结构与算法

(C++语言描述)

(第二版)

陈明◎编著

SHUJU JIEGOU
YU SUANFA



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社

图书在版编目 (CIP) 数据

数据结构与算法: C++语言描述/陈明编著. —2版. —北京: 北京师范大学出版社, 2017.11

新世纪高等学校规划教材. 计算机专业核心课程系列

ISBN 978-7-303-22812-6

I. ①数… II. ①陈… III. ①数据结构—高等学校—教材 ②算法分析—高等学校—教材 ③C++语言—程序设计—高等学校—教材 IV. ①TP311.12 ②TP301.6 ③TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 219416 号

营 销 中 心 电 话 010-62978190 62979006

北师大出版社科技与经管分社网 www.jswsbook.com

电 子 信 箱 jswsbook@163.com

出版发行: 北京师范大学出版社 www.bnup.com

北京新街口外大街 19 号

邮政编码: 100875

印 刷: 北京京师印务有限公司

经 销: 全国新华书店

开 本: 787 mm×1092 mm 1/16

印 张: 24

字 数: 548 千字

版 次: 2017 年 11 月第 1 版

印 次: 2017 年 11 月第 1 次印刷

定 价: 48.00 元

策划编辑: 赵洛育

责任编辑: 赵洛育

美术编辑: 刘 超

装帧设计: 刘 超

责任校对: 赵非非

责任印制: 赵非非

版权所有 侵权必究

反盗版、反侵权举报电话: 010-62978190

北京读者服务部电话: 010-62979006-8021

外埠邮购电话: 010-62978190

本书如有印装质量问题, 请与印制管理部联系调换。

印制管理部电话: 010-62979006-8006

内 容 简 介

本书较系统地介绍了各种典型的数据结构，主要包括：线性表、栈和队列、串、数组和广义表、树、图、查找、排序、递归和文件，为了加强对算法的理解，也介绍了算法分析方面的内容。本书叙述选材精炼、概念清楚、注重实用、逻辑性强，各章中所涉及的数据结构与算法都给出了 C++ 语言描述，并都附有大量丰富的习题，便于学生理解与掌握。

本书可作为高等院校计算机专业及相关专业的教材，也可作为计算机应用技术人员的参考书。

前 言

在非数值计算中,处理对象已从简单数值发展到具有一定结构的数据,这就需要讨论如何有效地组织计算机的存储,并在此基础上有效地实现对象间运算,数据结构就是研究与解决这些问题的重要基础。

作为一个 C++ 专业化程序员,必须能够熟练地选择和设计各种数据结构与算法,这是区分一个程序员水平的重要标志,缺少数据结构与算法的基础与功底,难以设计出具有专业水准的应用程序。

数据结构作为一门独立的课程,在国外是从 1968 年才开始设立的。1968 年美国唐纳德·克努特(Donald Ervin Knuth)教授开创了数据结构的最初体系,他所著的《计算机程序设计艺术第 1 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。数据结构在计算机科学中是一门综合性的专业基础课,数据结构是介于数学、计算机硬件和计算机软件三者之间的一门核心课程。数据结构这一门课的内容不仅是一般程序设计(特别是非数值性程序设计)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的重要基础。

计算机解决一个具体问题时,需要经过下列几个主要步骤:首先要从具体问题中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,最后以算法为核心设计出程序并进行测试、调整直至得到最终答案。

寻找数学模型的实质是通过分析问题,从中提取操作的对象,并找出这些操作对象之间含有的关系,然后用数学的语言加以描述。在用计算机处理数值计算问题所用的数学模型是用数学方程描述,所涉及的运算对象一般是简单的整型、实型和逻辑型数据,因此程序设计者的主要精力集中于程序设计技巧上,而不是数据的存储和组织上。但是,对于计算机应用领域是非数值型的计算问题,其数学模型无法用数学方程描述,而是用数据结构描述,解决此类问题的关键是设计出合适的数据结构,描述非数值型问题的数学模型是应用线性表、树、图等结构来描述。

计算机算法与数据的结构密切相关,算法依附于具体的数据结构,数据结构直接关系到算法的选择和效率。运算是由计算机来完成的,这就要设计相应的插入、删除和修改的算法。也就是说,数据结构还需要给出每种结构类型所定义的各种运算的算法。N.Wirth(沃斯)提出了“程序=数据结构+算法”的精辟描述。

数据结构课已经成为计算机科学与技术专业的一门必修的、重要的专业基础课,是计算机程序设计的重要理论与技术基础。通过对数据结构课的学习,不仅可以使同学们掌握数据结构的基本内容、典型算法和使用方法,而且能够训练学生应用数据结构和算法进行具体应用问题的程序设计。

本书分 12 章,介绍最常用的数据结构、各种数据结构的逻辑关系、在计算机中的存储表示以及在数据结构上的运算等。各章中所涉及的数据结构与算法都给出了 C++ 语言描述。

主要包括：算法分析、线性表、栈和队列、串、数组和广义表、树、图、查找、排序、递归和文件等内容。

本书在结构上呈积木式，注重实践应用，各种常用数据结构的介绍从实际出发，避免抽象的理论论述和复杂的公式推导，通过多个应用实例来说明各种数据结构和算法的设计方法。在典型的算法介绍中深入浅出、简洁明了。每章都设有小结和习题。通过这些习题的练习，不仅能加深对基本概念和定义的理解，而且通过上机，能够提高编程能力和程序调试能力。

本书可以作为大专院校计算机专业和相近专业的教材，也可作为从事计算机应用的工程技术人员参考书。

作者水平有限，书中不足之处在所难免，敬请读者批评指正。

陈 明

2017.2.1

目 录

第 1 章 概述.....	1
1.1 问题的提出.....	1
1.2 面向对象程序设计.....	2
1.2.1 面向对象方法.....	2
1.2.2 C++语言简介.....	3
1.3 数据结构的基本术语.....	4
1.4 抽象数据类型.....	6
1.4.1 抽象数据类型定义.....	6
1.4.2 抽象数据类型 ADT 描述规范.....	7
1.5 数据结构的组织形式.....	8
1.6 数据的逻辑结构.....	10
1.7 数据的存储结构.....	12
1.8 数据运算.....	13
1.9 数据的逻辑结构、存储结构及数据运算的关系.....	14
小结.....	14
习题 1.....	15
第 2 章 算法分析基础.....	16
2.1 算法相关概念.....	16
2.1.1 算法及其特性.....	16
2.1.2 算法的描述.....	17
2.2 算法分析的概念与算法复杂度.....	17
2.2.1 算法分析的概念.....	18
2.2.2 算法复杂度.....	20
2.3 最大连续子序列之和的问题.....	22
2.3.1 简单的 $O(N^3)$ 算法.....	23
2.3.2 改进的 $O(N^2)$ 算法.....	24
2.4 静态搜索问题.....	25
2.4.1 顺序搜索.....	25
2.4.2 二分搜索.....	26
小结.....	28
习题 2.....	28

第3章 线性表	31
3.1 线性表及其抽象数据类型	31
3.1.1 线性表及逻辑结构	31
3.1.2 线性表的抽象数据类型描述	35
3.2 线性表的顺序存储	36
3.2.1 顺序存储	36
3.2.2 顺序表的定义	37
3.2.3 顺序表类的实现	38
3.3 线性表的链式存储	41
3.3.1 线性链表的存储结构	41
3.3.2 线性链表类的定义	43
3.3.3 线性链表类的实现	44
3.3.4 循环链表	48
3.3.5 循环链表类的实现	49
3.3.6 双向链表	50
3.3.7 可利用空间表	52
3.3.8 表遍历器	53
3.4 线性表的顺序存储和链式存储的比较	55
3.5 链式存储结构的应用	56
3.5.1 约瑟夫问题	56
3.5.2 一元多项式求和	57
小结	61
习题3	61
第4章 栈和队列	64
4.1 栈	64
4.1.1 栈的定义	64
4.1.2 栈的顺序存储结构	67
4.1.3 栈的链式存储结构	71
4.1.4 顺序栈和链式栈的比较	73
4.2 栈的应用	74
4.2.1 迷宫问题	74
4.2.2 表达式求值	77
4.2.3 汉诺塔问题	80
4.2.4 数制转换	81
4.2.5 行编辑	82
4.3 队列	83
4.3.1 队列的定义	83

4.3.2	队列的顺序存储	86
4.3.3	队列的链式存储	94
4.3.4	顺序队列与链式队列的比较	97
4.3.5	优先队列	97
4.4	队列的应用	98
4.4.1	解决设备速度不匹配问题	98
4.4.2	舞伴问题	98
4.4.3	火车车厢重排	100
小结		102
习题 4		102
第 5 章 串		105
5.1	C++语言的字符和字符串	106
5.2	串的基本概念	107
5.3	串的存储结构	108
5.3.1	串的顺序存储结构	108
5.3.2	串的链式存储结构	109
5.3.3	串的索引存储结构	111
5.4	串的操作	111
5.4.1	常用的 C++字符串函数	111
5.4.2	串的抽象数据类型的描述	113
5.4.3	串的定义	114
5.4.4	部分成员函数的实现	115
5.5	串的基本运算与实现	117
5.5.1	串插入	117
5.5.2	串删除	119
5.6	模式匹配	120
5.6.1	模式匹配的 BF 算法	120
5.6.2	模式匹配的 KMP 算法	122
5.7	串在文本编辑中的应用	126
小结		128
习题 5		128
第 6 章 数组和广义表		129
6.1	C++数组的定义及抽象数据类型表示	129
6.1.1	C++数组的定义	129
6.1.2	数组的抽象数据类型表示	130
6.2	数组的顺序存储结构	131
6.3	矩阵的压缩存储	132

6.3.1 特殊矩阵的压缩存储	133
6.3.2 稀疏矩阵的压缩存储	136
6.4 广义表的概念	142
6.5 广义表的存储结构表示	144
6.6 广义表的运算	146
小结	152
习题 6	152
第 7 章 树	155
7.1 树的相关术语	156
7.1.1 树的定义	156
7.1.2 树的表示形式	157
7.1.3 树的元素	157
7.1.4 树的基本操作	159
7.1.5 一个树的接口	159
7.1.6 树的基本算法	160
7.2 二叉树	161
7.2.1 二叉树的定义	162
7.2.2 二叉树的性质	163
7.2.3 二叉树的接口	166
7.2.4 二叉树的存储结构	166
7.2.5 二叉树的遍历	174
7.2.6 二叉树遍历的应用	176
7.3 线索二叉树	177
7.3.1 线索二叉树的类定义	177
7.3.2 中序线索二叉树	182
7.4 树、森林和二叉树的关系	185
7.4.1 树的存储结构	185
7.4.2 森林与二叉树的转换	187
7.4.3 树和森林的遍历	190
7.5 哈夫曼树及其应用	192
7.5.1 哈夫曼树的定义	192
7.5.2 哈夫曼树的构造	193
7.5.3 哈夫曼树在编码问题中的应用	196
小结	197
习题 7	198
第 8 章 图	200
8.1 图的定义与抽象数据类型	200

8.1.1 图的定义.....	200
8.1.2 图的抽象数据类型.....	205
8.2 图的存储结构.....	206
8.2.1 邻接矩阵表示法.....	207
8.2.2 邻接表.....	212
8.2.3 十字链表.....	219
8.2.4 邻接多重表.....	220
8.3 图的遍历.....	221
8.3.1 深度优先搜索.....	222
8.3.2 广度优先搜索.....	223
8.3.3 欧拉回路.....	225
8.4 图的连通性.....	227
8.4.1 连通分量.....	227
8.4.2 重连通分量.....	229
8.5 生成树.....	229
8.5.1 普里姆 (Prim) 算法.....	231
8.5.2 克鲁斯卡尔 (Kruskal) 算法.....	234
8.6 最短路径.....	236
8.6.1 单源最短路径.....	237
8.6.2 求每一对顶点之间的最短路径.....	239
8.7 拓扑排序.....	240
8.8 关键路径.....	245
小结.....	250
习题 8.....	251
第 9 章 查找.....	253
9.1 基本概念.....	253
9.2 线性表的查找.....	254
9.2.1 顺序查找.....	255
9.2.2 折半查找.....	256
9.2.3 索引查找.....	259
9.2.4 分块查找.....	262
9.3 树表查找.....	265
9.3.1 二叉查找树.....	265
9.3.2 平衡二叉树.....	271
9.3.3 B-树.....	276
9.4 哈希表的查找.....	278
9.4.1 哈希表.....	278

9.4.2	构造哈希表的基本方法	280
9.4.3	解决冲突的方法	282
9.4.4	哈希表的查找方法	284
9.5	各种查找方法的比较	285
小结		286
习题 9		286
第 10 章	排序	289
10.1	基本概念及分类	289
10.2	内部排序	293
10.2.1	插入排序	293
10.2.2	交换排序	297
10.2.3	选择排序	301
10.2.4	归并排序	311
10.2.5	基数排序	314
10.3	内部排序方法比较	318
10.4	外部排序简介	320
小结		320
习题 10		320
第 11 章	递归	324
11.1	递归的定义	324
11.2	常见递归问题	325
11.2.1	汉诺塔问题	325
11.2.2	八皇后问题	327
11.2.3	表达式树	329
11.3	递归的实现	331
11.4	消除递归	334
11.4.1	尾递归和单向递归的消除	335
11.4.2	用栈模拟系统运行时的栈	336
11.5	递归的评估	339
小结		340
习题 11		340
第 12 章	文件	342
12.1	外存储器的介绍	342
12.2	磁盘	343
12.3	有关文件的概念	344
12.3.1	文件及其类别	345

12.3.2 文件的操作	346
12.4 文件的组织	348
12.4.1 顺序文件	348
12.4.2 索引文件	350
12.4.3 散列文件	355
12.4.4 多关键字文件	357
12.5 外部排序	359
12.5.1 外部排序的简单方法	359
12.5.2 两路归并	360
12.5.3 多路归并	363
12.6 文件的索引结构	363
12.6.1 索引向量	364
12.6.2 树形索引结构	364
小结	365
习题 12	365

第 1 章 概 述



本章主要内容



1.1 问题的提出

在计算机发展的初期，人们使用计算机主要是处理数值的计算问题，程序设计人员也主要把精力集中在程序设计的技巧上，但随着计算机应用领域的扩大和软硬件的发展，计算机对信息的处理加工已从单一的数值计算发展到大量地解决非数值问题，其加工处理的信息也由简单的数值发展到字符、图像、声音等具有复杂结构的数据。而数据结构就是随着计算机的产生和发展而发展起来的一门较新的计算机学科。

非数值问题的数据之间的相互关系一般无法完全用数学方程式加以描述，并且数据的表示方法和组织形式直接关系到程序对数据的处理效率，而系统程序和许多应用程序的规模很大，结构复杂，这时人们考虑问题的关键已不再是分析数学和计算方法，而是放在是否能设计出合适的数据结构，有效地解决问题上。

计算机科学是一门研究用计算机进行信息表示和处理的科学。这里面涉及两个问题：信息的表示和信息的处理。而信息的表示和组成又直接关系到处理信息的程序的效率。随着计算机的普及，信息量的增加，信息范围的拓宽，使许多系统程序和应用程序的规模很大，结构又相当复杂。这就要求人们对计算机程序加工的对象进行系统的研究，即研究数据的特性以及数据之间存在的关系，而数据结构（Data Structure）正是描述数据的特性以及数据之间存在的关系的一门课程。

例 1-1 电话号码查询问题。

编一个查询某个城市或单位的私人电话号码的程序。要求对任意给出的一个姓名，若

该人有电话号码，则迅速找到其电话号码；否则指出该人没有电话号码。

要解此问题，首先构造一张电话号码登记表。表中每个结点存放两个数据项：姓名和电话号码。

要写出好的查找算法，取决于这张表的结构及存储方式。最简单的方式是将表中结点按顺序地存储在计算机中。查找时从头开始依次查对姓名，直到找出正确的姓名或是找遍整个表均没有找到为止。这种查找算法对于一个不大的单位或许是可行的，但对一个有成千上万私人电话的城市就不实用了。若这张表是按姓氏排列的，则可另造一张姓氏索引表，采用如图 1-1 所示的存储结构。那么查找过程是先在索引表中查对姓氏，然后根据索引表中的姓氏到电话号码登记表中核查姓名，这样查找登记表时就无须查找其他姓氏的名字了。因此，在这种新的结构上产生的查找算法就更为有效。

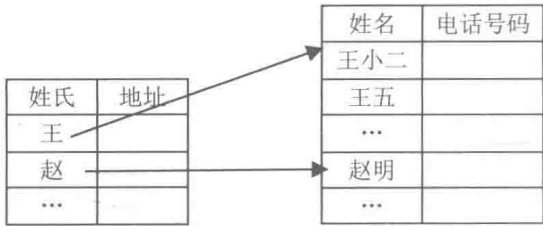


图 1-1 带索引表的电话号码查找

数据结构是计算机专业的核心课程之一，在众多的计算机系统软件和应用软件中都要用到各种数据结构。可以这样说，数据结构不仅是一般程序设计的基础，而且是实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的基础。因此，仅掌握几种计算机语言是难以应付众多复杂的研究课题的，要想有效地使用计算机，还必须学习数据结构的知识。

瑞士计算机科学家 N.Wirth 教授曾提出这样一个等式：算法+数据结构=程序，这个等式形象地描述了算法、数据结构和程序之间的关系，这里的数据结构指的是数据的逻辑结构和存储结构，而算法就是对数据运算的描述。由此可见，程序设计的实质就是对实际问题选取一种好的数据结构，加之设计一个好的算法，而且好的算法很大程度上取决于描述实际问题的数据结构。对于一个程序设计人员来说，学好数据结构是十分重要的，那什么是数据结构呢，下面一节就是对数据结构和一些相关概念的解释。

1.2 面向对象程序设计

1.2.1 面向对象方法

1. 面向对象

简单地讲，面向对象是把一组相互无联系的对象有效地集成在一起的软件，而这些对象都是将数据结构和行为封装在一起的。这与传统的程序设计将数据结构与行为分离完全

不同。

2. 面向对象方法的概念

面向对象方法是基于客观世界的对象模型的软件开发方法。当设计和实现一个客观系统时，如能在满足需求的条件下，把系统设计成由一些相对固定的部分组成的最小集合，则这个设计就是优秀的，而这些相对固定的部分就被看成是一些不同的对象。面向对象方法就是要建立与语言无关的对象模型，并围绕这些对象进行系统设计。

3. 面向对象方法的基本特征

(1) 对象是数据和操作的封装体，较好地实现了数据的抽象。

(2) 面向对象方法的继承性体现了概念分离抽象。下层对象继承上层对象的属性和操作，因而便于软件的演化和扩充。

(3) 面向对象方法具有信息隐藏性。对象将其实现的细节隐藏在内部，所以无论是对对象功能的完善扩充，还是对象实现的修改，都不会对外界产生影响。这就保证了面向对象软件的可构造性和易维护性。

(4) 面向对象方法用消息将对象动态链接在一起。面向对象方法采用了灵活的消息传递方式，从而便于在概念上体现并行和分布式结构。

1.2.2 C++语言简介

1. C++语言的重要特征

- 类和数据封装
- 构造函数和析构函数
- 私有、公有和受保护
- 对象和消息
- 友元函数
- 类中运算符和函数名重载
- 派生类
- 虚函数
- 流库

2. C++语言的基本概念

(1) 类：类是一种数据结构，它把数据和处理数据的过程封装在一起。类和普通数据类型的主要区别，在于类可以拥有成员数据和成员函数。

(2) 构造函数：构造函数是一种特殊的成员函数，用于在内存中建立具体的对象。构造函数必须初始化成员变量，申请必要的内存空间，并将内存转化成对象。一个类可以同时拥有几个带有不同参数的构造函数。

(3) 析构函数：析构函数与构造函数正好相反，它是一种当对象被取消时才被调用的

特殊的成员函数。一个类只能有一个析构函数。析构函数和构造函数一样都没有返回类型和返回值。

(4) 私有、公有和受保护：每个类的成员变量和成员函数都有 3 种访问权限，即 `private`、`public` 和 `protected`。声明为 `private` 的成员变量和成员函数只能被该类的成员函数访问；声明为 `public` 的成员变量和成员函数可以被其他类的成员函数访问；声明为 `protected` 的成员变量和成员函数不能被其他类的成员函数访问，但子类可以访问父类中声明为 `protected` 的成员函数。

(5) 对象和消息：对象是用发送的消息操作的，对象响应消息。每个消息都用类中相应的方法给出。

(6) 重载：重载包含运算符重载和函数重载。运算符重载是指对已存在的运算符赋予新的含义。函数重载是指相同的函数名称，如果参数个数不同，或者是参数个数相同但类型不同，函数具有不同的功能。重载函数的意义在于它能用同一个函数名访问一组相关的函数。

(7) 继承：继承是在已有类的基础上定义新类的机制。继承又可以分为单继承和多重继承。单继承是指派生类继承自一个基类，而多重继承则是派生类继承自两个或两个以上的基类。

(8) 多态性：多态性是指程序在执行过程中可以保持多种不同类型的值。具有多态性的函数可以对多种不同类型的对象进行操作。

(9) 派生类：派生类是由基类派生出来的，它继承父类所有或部分方法。除此之外，派生类还可以定义自己的方法。

(10) 虚函数：虚函数是在函数前加上关键字 `virtual`。虚函数支持多态性。当虚函数被调用时，适合于当前对象的函数被执行，尽管当前对象的类型直到运行时才能够知道。

以上是 C++ 语言中的一些基本概念，这里只是简单地介绍，具体的实现可以查阅相关的面向对象程序设计语言的书籍。

1.3 数据结构的基本术语

为了更好地理解数据结构这个概念，先给出数据结构中的一些常用基本名词和术语的解释。

1. 数据

数据是信息的载体，它是描述客观事物的数、字符以及所有能输入到计算机中被计算机程序识别、加工处理的信息的集合。数据不只是通常意义下的整数和实数等。随着计算机的广泛应用，数据的范畴也随之扩大，计算机可以处理的字符串、图像、声音等都可以被称为数据，所以不能只是泛泛地理解数据这个概念。下面进一步解释一下数据的定义。如表 1-1 所示，例如，张风的英语成绩为 92 分，92 就是该同学的成绩数据。