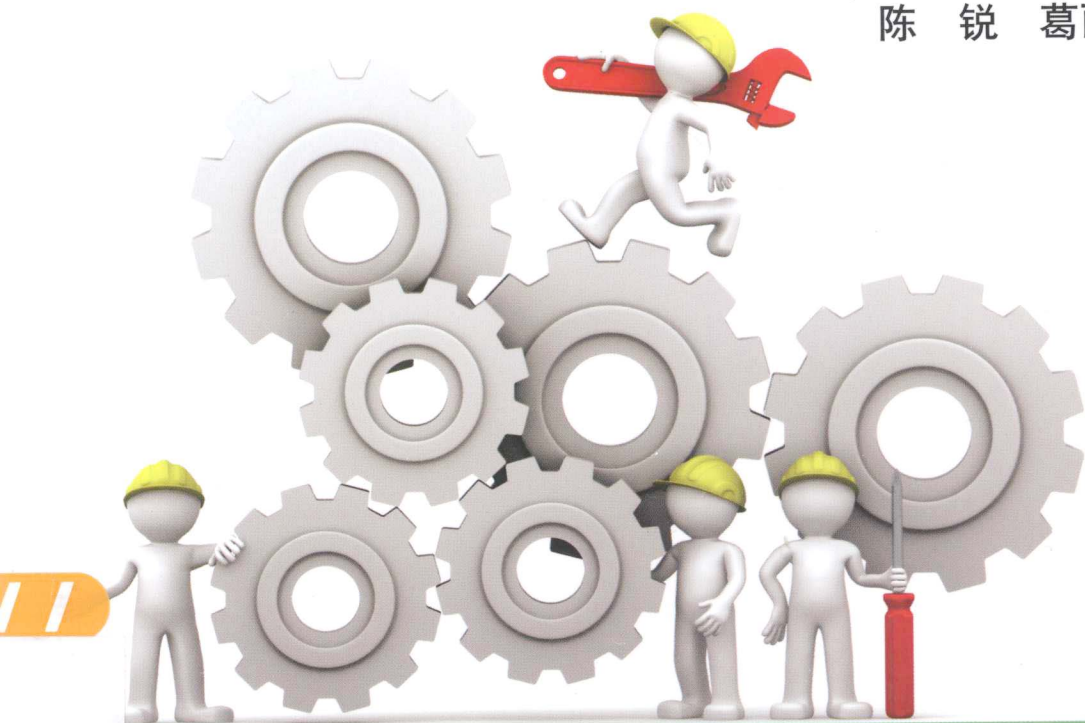


跟我学 数据结构

陈 锐 葛丽萍 编著



涵盖数据结构中的所有知识点。以浅显易懂的语言讲解每一个专业术语

- 以类比的方式帮助读者理解抽象的概念
- 提供70多个典型实例帮助读者理解数据结构和算法设计思想
- 实例源于多年的考研试题研究与实践成果，案例典型且涵盖知识点丰富



清华大学出版社

013064673

TP311.12

266

跟我学数据结构

陈 锐 葛丽萍 编著



清华大学出版社



北航

C1672346

TP311.12
266

013084873

内 容 简 介

数据结构是计算机专业的基础和核心课程。本书内容全面, 语言通俗易懂, 案例典型、丰富, 结构清晰, 重点难点突出, 所有算法都有完整程序, 能直接运行。本书内容包括数据结构概述、常用的 C 程序开发环境、线性表、栈、队列、串、数组、广义表、树、图、查找、排序。

本书可作为从事计算机软件开发、准备考取计算机专业研究生和参加软考的人员学习数据结构与算法的参考书, 也可以作为计算机及相关专业的数据结构课程教材。

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

跟我学数据结构/陈锐, 葛丽萍编著. —北京: 清华大学出版社, 2013
ISBN 978-7-302-33009-7

I. ①跟… II. ①陈… ②葛… III. ①数据结构 IV. ①TP311.12

中国版本图书馆 CIP 数据核字(2013)第 145936 号

责任编辑: 张彦青

封面设计: 杨玉兰

责任校对: 周剑云

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>, 010-62791865

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 31.75

字 数: 775 千字

版 次: 2013 年 9 月第 1 版

印 次: 2013 年 9 月第 1 次印刷

印 数: 1~3000

定 价: 58.00 元

产品编号: 049947-01

前言

数据结构是计算机专业基础课，在所有计算机课程中占据举足轻重的地位，也是一门不太容易掌握的课程。但是，不要因此而气馁，本书将采用通俗的语言，教你掌握好数据结构的相关知识。

如果你的学习目的是将来成为一名优秀的程序员，向微软、谷歌、百度的工程师们看齐，那么你应该努力学好数据结构知识，不仅要看懂书中的程序和算法，还要完成课后的习题，并上机实践。

如果你的学习目的是为了考取计算机专业的研究生，数据结构作为计算机专业考研的重头戏，也是今后继续深造的必备基础，你需要认真研读这本书，真正领会每一个算法思想，做到给出任何一个题目，都能自己动手写出算法。

当然，如果你仅仅是为了混学分，顺利通过考试，那么也需要认真看完这本书，这本书可以作为你学习遇到困难时的参考书，方便随时查阅，可在本书中找到你需要的答案。

在写作本书之前，我曾写过几本关于数据结构方面的著作，得到了一些读者的厚爱，收到了热心读者的来信，他们提出了宝贵的建议。本书吸取了过去的一些经验，修订了其中的错误，努力写得更好，希望有更多的读者喜欢。本书适用于在读的计算机专业学生、准备考取计算机专业研究生的人员和从事教学科学研究的人员阅读。但如果您觉得这本书并不适合您，请将本书放回书架比较显眼的位置，我们在此表示感谢。

本书是一本难得的内容完整、语言通俗、案例丰富的数据结构自学图书和教材。本书致力于将数据结构这个原本抽象的东西尽可能地通俗化，让每位希望掌握数据结构知识的朋友都能尽快轻松地掌握它，因此本书在表述方面采用了通俗的语言，并选取了丰富的案例，以满足读者的需要。

本书全面地介绍数据结构的基本知识，通过理论和实践并重的方式，站在初学者的角度，从最基础的知识开始，由浅入深，对每一个概念都通过通俗的语言进行讲解，对每一个抽象的概念，对都拿现实生活中的例子进行类比，以方便读者理解和掌握。另外，本书案例丰富、典型，所有算法都直接利用 C 语言描述，所有程序都可以直接运行。本书内容全面，不仅详细介绍了 C 语言的基础知识，还涉及 C 语言相关的高级技术和理论知识，是一本难得的技术参考书和自学教材，主要内容包括数据结构概述、常用的程序开发环境、线性表、栈、队列、串、数组、广义表、树与二叉树、图、查找、排序。

通过学习本书，您会体验到学习数据结构时从未有过的简单易学，本书将帮您轻松掌握数据结构中的每一个知识点，攻克数据结构知识堡垒中的任何一个重点和难点。

1. 本书的特点

(1) 内容全面，讲解详细：为了方便读者学习，本书首先对数据结构课程的目标和描述方式进行介绍，并对算法使用的语言——C 语言的重点和难点进行复习。本书覆盖了数据结构中线性表、树和图的所有知识点，对于每一种数据结构，都使用所有可能的逻辑结构

和存储结构进行描述,并对算法尽量采用多种实现方式,如递归和非递归、顺序存储和链式存储,从而使读者对算法的理解更加深刻。

(2) 层次清晰,结构合理:本书将数据结构分篇、章、节和小节划分知识点,将知识点细化,易于读者理解。每一个知识点单独作为一个小节,专门讲解。在知识点的讲解过程中,循序渐进,由浅入深,先引出概念,再用例子说明,然后是算法描述,最后是具体程序实现。这样的层次十分易于读者理解和消化。

(3) 结合图表,语言通俗:在每个概念提出后,都结合图表和例子以方便读者理解。在语言的叙述上,普遍采用短句子、易于理解的语言,而避免使用复杂句子和晦涩难懂的语言。通过以上方式的描述,读者可以更加容易和轻松地学习数据结构。

(4) 例子典型,深入剖析:在讲解每一个算法时,结合具体例子进行剖析。在例子的选取上,结合历年考研试题,选取的例子涵盖知识点比较全面,具有代表性。在每一章的最后或比较大的知识点后面,都给出了一个完整的程序,给出程序的同时,还对算法通过图进行具体讲解,深入分析,并在程序的最后给出运行结果。读者在学习的过程中,可以结合例子和运行结果以验证算法的正确性。

(5) 配有习题,巩固知识:从第2章开始,在每一章的最后,都有一个小结,对该章的知识点进行总结。为了让读者熟练编写算法,本书在每一章的最后都配有一定数量的实践题目,在学习了每一章的内容之后,可以通过这些习题试着编写算法,以巩固该章的学习内容。在本书配套的下载资料中,提供了每一个例子的程序代码和课后习题代码。

2. 本书的内容

第1章:如果读者刚接触数据结构,该章将从什么是数据结构讲起,介绍本书的学习目标、学习方法和学习内容,作者还将现身说法,告诉读者如何学好数据结构知识。

第2章:对本书的描述语言和使用工具进行介绍。该章主要介绍C语言的开发环境,然后复习C语言中的重点和难点——指针、数组、函数、递归和结构体。通过对该章内容的学习,读者在以后数据结构的学习过程中将会得心应手。

第3章:主要介绍线性表。首先讲解线性表的逻辑结构,然后介绍线性表的两种常用存储结构,并讲解各种链表结构,包括静态链表,在每一节均给出算法的具体应用。通过对该章内容的学习,读者将掌握顺序表和各种链表的操作。

第4章:主要介绍一种特殊的线性表——栈。首先介绍栈的定义,然后介绍栈的应用及栈与递归的关系、转化。通过对该章内容的学习,读者将学会栈的使用并深入理解递归和栈的知识。

第5章:主要介绍另一种特殊的线性表——队列。首先介绍队列的概念,然后介绍顺序队列、循环队列和链式队列,并给出各种队列的实现算法。该章在最后结合具体的例子分析队列的具体使用。

第6章:主要介绍另一种特殊的线性表——串。首先介绍串的概念,然后介绍串的各种存储表示,并介绍串的模式匹配算法。通过串的模式匹配可以提高求子串的效率。

第7章:主要介绍数组。首先介绍数组的概念,然后介绍数组(矩阵)的顺序存储、链式存储及矩阵的运算,最后介绍几种特殊的矩阵。通过对该章内容的学习,读者将掌握矩阵的一些算法操作。

第8章：主要介绍广义表。首先介绍广义表的概念，然后介绍广义表的两种存储方式，最后给出广义表的操作实现。

第9章：主要介绍一种非线性数据结构——树和二叉树。首先介绍树和二叉树的概念，然后介绍树和二叉树的存储表示、二叉树的性质、二叉树的遍历和线索化、树和森林与二叉树的转换及哈夫曼树。该章在讲解这些知识点时，均给出具体例子以增强对这些知识的理解。在该章的最后，专门给出树的具体应用。

第10章：主要介绍另一种非线性数据结构——图。首先介绍图的概念和存储结构，然后介绍图的遍历、最小生成树、拓扑排序、关键路径及最短路径。在讲解这些知识点时，都给出相应的算法和例子，以加强对知识点的理解。

第11章：主要介绍一种数据结构的常用技术——查找。查找是数据结构非数值运算中非常常用的技术，该章首先介绍查找的概念，然后介绍各种查找算法，并结合具体实例进行详细的讲解，并给出完整程序。通过对该章内容的学习，读者将掌握程序设计中非常重要的查找技术。

第12章：主要介绍另一种数据结构的常用技术——排序。排序是数据结构中最为常用的技术，该章首先介绍排序的相关概念，然后介绍多种排序技术，并结合实例讲解这些算法的实现，在每一节都给出完整的程序。通过对该章内容的学习，读者将掌握程序设计中最为常用的排序技术。

由于作者水平有限，书中难免存在一些不足之处，恳请读者批评指正。读者可以通过电子邮箱 nwuchenrui@126.com 与作者联系。

3. 适合的读者

本书适合下列读者阅读和学习使用：

- 大中专院校的学生。
- 准备考取计算机专业研究生的人员。
- 准备参加软考的人员。
- 软件开发人员。
- 计算机相关的科研工作者。

3. 致谢

感谢我的导师张蕾教授，她丰富的知识储备及敏锐的洞察力极大地影响了我的学习态度和认识能力，使我在职业生涯中受益，也为本书的编写奠定了良好的基础。

感谢我的家人，是因为有他们默默的付出和鼓励，我才能顺利地做好各项工作。

最后特别感谢温县教育局及所有支持我写作的朋友们！

陈 锐

目 录

第1章 概述.....1

1.1 数据结构的基本概念.....2

1.2 抽象数据类型.....5

1.2.1 抽象数据类型的定义.....5

1.2.2 抽象数据类型的描述.....6

1.3 算法的特性与算法的描述.....7

1.3.1 算法的定义.....7

1.3.2 算法的特性.....7

1.3.3 算法的描述.....8

1.4 算法分析.....9

1.4.1 算法设计的要求.....9

1.4.2 算法效率评价.....10

1.4.3 时间复杂度.....11

1.4.4 空间复杂度.....13

1.5 如何学好数据结构.....14

1.5.1 数据结构课程的地位.....14

1.5.2 数据结构课程的重要性.....14

1.5.3 如何学好数据结构.....15

第2章 C语言基础.....17

2.1 开发环境介绍.....18

2.1.1 Turbo C 2.0 开发环境介绍.....18

2.1.2 Visual C++ 6.0 开发环境介绍.....20

2.2 递归与非递归.....24

2.2.1 函数的递归调用.....24

2.2.2 递归函数应用举例.....25

2.2.3 一般递归转化为非递归 (使用迭代).....27

2.3 指针.....28

2.3.1 指针变量.....28

2.3.2 指针变量的引用.....29

2.3.3 指针与数组.....30

2.3.4 函数指针与指针函数.....35

2.4 参数传递.....42

2.4.1 传值调用.....42

2.4.2 传地址调用.....43

2.5 结构体与共用体.....46

2.5.1 结构体的定义.....46

2.5.2 指向结构体的指针.....48

2.5.3 共用体及应用.....49

2.6 动态内存分配与释放.....50

2.6.1 内存动态分配与释放.....50

2.6.2 链表.....51

2.7 小结.....56

2.8 习题.....57

第3章 线性表.....59

3.1 线性表的概念及抽象数据类型.....60

3.1.1 线性表的定义.....60

3.1.2 线性表的抽象数据类型.....61

3.2 线性表的顺序表示与实现.....62

3.2.1 线性表的顺序存储结构.....62

3.2.2 顺序表的基本运算.....63

3.2.3 顺序表基本运算的 算法分析.....66

3.3 顺序表的应用举例.....67

3.4 线性表的链式表示与实现.....72

3.4.1 单链表的存储结构.....72

3.4.2 单链表上的基本运算.....74

3.5 单链表应用举例.....79

3.6 循环单链表.....87

3.6.1 循环链表的链式存储.....87

3.6.2 循环单链表的应用.....88

3.7 双向链表.....93

3.7.1 双向链表的存储结构.....93

3.7.2 双向链表的插入操作和 删除操作.....94



3.8	双向链表的应用	96
3.9	静态链表	99
3.9.1	静态链表的存储结构	99
3.9.2	静态链表的实现	100
3.9.3	静态链表的应用	102
3.10	各种线性表的操作	104
3.11	一元多项式的表示与相乘	111
3.11.1	一元多项式的表示	112
3.11.2	一元多项式相乘	113
3.12	小结	117
3.13	习题	118
第4章	栈	121
4.1	栈的表示与实现	122
4.1.1	栈的定义	122
4.1.2	栈的抽象数据类型	123
4.2	栈的顺序表示与实现	124
4.2.1	栈的顺序存储结构	124
4.2.2	顺序栈的基本运算	125
4.2.3	共享栈的问题	127
4.3	栈的应用举例	129
4.4	栈的链式表示与实现	132
4.4.1	栈的存储结构	133
4.4.2	栈的基本运算	133
4.4.3	链栈的应用	136
4.5	栈的应用举例	137
4.5.1	数制转换	137
4.5.2	括号配对	139
4.5.3	行编辑程序	141
4.6	栈与递归的实现	143
4.6.1	递归	143
4.6.2	消除递归	146
4.7	栈的应用举例	152
4.7.1	表达式的转换与运算	152
4.7.2	表达式的运算举例	154
4.8	小结	158
4.9	习题	159

第5章	队列	161
5.1	队列的定义	162
5.1.1	队列的定义	162
5.1.2	队列的抽象数据类型	162
5.2	队列的顺序存储及实现	163
5.2.1	顺序队列的表示	163
5.2.2	顺序队列的“假溢出”	166
5.2.3	顺序循环队列的表示	167
5.2.4	顺序循环队列的实现	168
5.2.5	顺序循环队列实例	170
5.3	队列的链式存储及实现	173
5.3.1	链式队列的表示	173
5.3.2	链式队列的实现	175
5.3.3	链式队列实例	177
5.4	双端队列	181
5.4.1	双端队列的定义	181
5.4.2	双端队列的应用	181
5.5	队列在杨辉三角中的应用	184
5.5.1	杨辉三角	184
5.5.2	杨辉三角的队列构造	185
5.5.3	杨辉三角队列的实现	185
5.6	小结	189
5.7	习题	190
第6章	串	191
6.1	串	192
6.1.1	串的定义	192
6.1.2	串的抽象数据类型	192
6.2	串的顺序表示与实现	195
6.2.1	串的顺序存储结构	195
6.2.2	串的基本运算	196
6.3	串的应用举例	201
6.4	串的堆分配表示与实现	202
6.4.1	堆分配的存储结构	202
6.4.2	堆串的基本运算	203
6.5	堆串的应用举例	209
6.6	串的链式存储表示与实现	210
6.6.1	串的链式存储结构	210
6.6.2	链串的基本运算	212

6.7 链串的应用举例.....	217
6.8 串的模式匹配.....	219
6.8.1 经典的模式匹配算法	
Brute-Force	219
6.8.2 KMP 算法	220
6.8.3 模式匹配应用举例.....	226
6.9 小结.....	230
6.10 习题.....	230
第7章 数组	233
7.1 数组.....	234
7.1.1 数组的定义.....	234
7.1.2 数组的抽象数据类型.....	235
7.2 数组的顺序表示与实现.....	235
7.2.1 数组的顺序存储结构.....	236
7.2.2 数组的基本运算.....	237
7.2.3 数组的应用举例.....	239
7.3 特殊矩阵的压缩存储.....	241
7.3.1 对称矩阵的压缩存储.....	241
7.3.2 三角矩阵的压缩存储.....	242
7.3.3 对角矩阵的压缩存储.....	243
7.4 稀疏矩阵的压缩存储.....	243
7.4.1 稀疏矩阵的定义.....	244
7.4.2 稀疏矩阵抽象数据类型.....	244
7.4.3 稀疏矩阵的三元组表示.....	245
7.4.4 稀疏矩阵的三元组实现.....	246
7.5 稀疏矩阵的应用举例.....	252
7.5.1 稀疏矩阵相乘的三元组	
表示.....	252
7.5.2 稀疏矩阵的相乘三元组	
实现.....	254
7.6 稀疏矩阵的十字链表表示与实现.....	257
7.6.1 稀疏矩阵的十字链表表示.....	257
7.6.2 十字链表的实现.....	258
7.7 稀疏矩阵的十字链表实现应用	
举例.....	261
7.8 小结.....	266
7.9 习题.....	267

第8章 广义表..... 269

8.1 广义表.....	270
8.1.1 广义表的定义.....	270
8.1.2 广义表的抽象数据类型.....	271
8.2 广义表的头尾链表表示与实现.....	271
8.2.1 广义表的头尾链表存储	
结构.....	272
8.2.2 广义表的基本运算.....	273
8.2.3 广义表的应用举例.....	275
8.3 广义表的扩展线性链表表示与	
实现.....	278
8.3.1 广义表的扩展线性链表	
存储.....	278
8.3.2 广义表的基本运算.....	279
8.3.3 采用扩展线性链表存储结构的	
广义表应用举例.....	282
8.4 小结.....	284
8.5 习题.....	285

第9章 树..... 287

9.1 树.....	288
9.1.1 树的定义.....	288
9.1.2 树的逻辑表示.....	289
9.1.3 树的抽象数据类型.....	290
9.2 二叉树.....	291
9.2.1 二叉树的定义.....	291
9.2.2 二叉树的性质.....	293
9.2.3 二叉树的抽象数据类型.....	294
9.3 二叉树的存储表示与实现.....	295
9.3.1 二叉树的顺序存储.....	296
9.3.2 二叉树的链式存储.....	296
9.3.3 二叉树的基本运算.....	297
9.4 二叉树的遍历.....	301
9.4.1 二叉树遍历的定义.....	301
9.4.2 二叉树的先序遍历.....	301
9.4.3 二叉树的中序遍历.....	303
9.4.4 二叉树的后序遍历.....	305
9.5 二叉树的遍历的应用举例.....	307
9.5.1 二叉树的创建.....	308



9.5.2	二叉树的输出	311
9.5.3	二叉树的计数	315
9.6	二叉树的线索化	318
9.6.1	二叉树的线索化定义	318
9.6.2	二叉树的线索化	319
9.6.3	线索二叉树的遍历	321
9.6.4	线索二叉树的应用举例	323
9.7	树、森林与二叉树	326
9.7.1	树的存储结构	326
9.7.2	树转换为二叉树	328
9.7.3	森林转换为二叉树	330
9.7.4	二叉树转换为树和森林	330
9.7.5	树和森林的遍历	331
9.8	哈夫曼树	332
9.8.1	哈夫曼树的定义	332
9.8.2	哈夫曼编码	334
9.8.3	哈夫曼编码算法的实现	334
9.9	树与二叉树的应用举例	340
9.9.1	相似二叉树	340
9.9.2	由先序和中序、中序和后序 确定二叉树	341
9.9.3	树的孩子兄弟链表应用 举例	347
9.10	小结	350
9.11	习题	350

第10章 图

10.1	图的定义与相关概念	354
10.1.1	图的定义	354
10.1.2	图的相关概念	354
10.1.3	图的抽象数据类型	357
10.2	图的存储结构	358
10.2.1	邻接矩阵表示法	358
10.2.2	邻接表表示法	360
10.2.3	十字链表表示法	361
10.2.4	邻接多重链表表示法	362
10.3	图的应用举例	364
10.3.1	采用邻接矩阵创建图	364
10.3.2	采用邻接表创建图	367

10.4	图的遍历	370
10.4.1	图的深度优先遍历	370
10.4.2	图的广度优先遍历	373
10.4.3	图的遍历应用举例	375
10.5	图的连通性问题	377
10.5.1	无向图的连通分量与 生成树	378
10.5.2	最小生成树	379
10.6	有向无环图	384
10.6.1	AOV网与拓扑排序	384
10.6.2	AOE网与关键路径	387
10.6.3	关键路径应用举例	392
10.7	最短路径	396
10.7.1	从某个顶点到其余各顶点的 最短路径	397
10.7.2	每一对顶点之间的 最短路径	402
10.8	图的应用举例	406
10.8.1	距离某个顶点的最短路径 长度为k的所有顶点	406
10.8.2	求图中顶点u到顶点v的 简单路径	409
10.9	小结	411
10.10	习题	412

第11章 查找

11.1	查找的基本概念	414
11.2	静态查找	414
11.2.1	顺序表的查找	415
11.2.2	有序顺序表的查找	416
11.2.3	索引顺序表的查找	418
11.2.4	静态查找应用举例	420
11.3	动态查找	422
11.3.1	二叉排序树	423
11.3.2	平衡二叉树	430
11.4	B-树与B+树	438
11.4.1	B-树	438
11.4.2	B+树	446
11.5	哈希表	447

11.5.1 哈希表的定义.....	447	12.3.3 选择排序应用举例.....	470
11.5.2 哈希函数的构造方法.....	448	12.4 交换排序.....	471
11.5.3 处理冲突的方法.....	449	12.4.1 冒泡排序.....	471
11.5.4 哈希表应用举例.....	451	12.4.2 快速排序.....	473
11.6 小结.....	454	12.4.3 交换排序应用举例.....	475
11.7 习题.....	455	12.5 归并排序.....	479
第 12 章 排序	457	12.5.1 归并排序算法.....	479
12.1 排序的基本概念.....	458	12.5.2 归并排序应用举例.....	481
12.2 插入排序.....	459	12.6 基数排序.....	482
12.2.1 直接插入排序.....	459	12.6.1 基数排序算法.....	483
12.2.2 折半插入排序.....	460	12.6.2 基数排序应用举例.....	486
12.2.3 希尔排序.....	461	12.7 各种排序算法的比较.....	489
12.2.4 插入排序应用举例.....	462	12.8 排序算法应用举例.....	490
12.3 选择排序.....	464	12.9 小结.....	494
12.3.1 简单选择排序.....	464	12.10 习题.....	495
12.3.2 堆排序.....	465	参考文献	496



第 1 章 概述

数据结构是计算机专业的一门基础课程，是我们以后学习计算机软件和计算机硬件的重要基础。它主要研究数据在计算机中的存储表示和对数据的处理方法。数据结构把数据划分为集合、线性表、树和图 4 种类型，而后 3 种是数据结构研究的重点。本章主要学习数据结构的基本概念、抽象数据类型及描述、算法的特性及要求。

通过阅读本章，您可以：

- 了解数据结构的基本概念。
- 掌握抽象数据类型的描述方法。
- 理解数据结构的逻辑结构和存储结构。
- 了解算法的特性。
- 掌握算法的描述方法及算法分析方法。



1.1 数据结构的基本概念

本节主要学习数据结构的一些基本概念和术语, 主要包括数据、数据元素、数据对象、数据结构、逻辑结构、存储结构及数据类型。

1. 数据

数据(Data)是描述客观事物的符号, 是能被计算机识别并能输入到计算机中处理的符号集合。数据不仅包括整型、实型等数值类型, 还包括字符及声音、图像、视频等非数值数据。例如, 王鹏的身高是 172cm, 这里“王鹏”是对一个人姓名的描述数据, “172cm”是关于身高的描述数据。一张照片是图像数据, 一部电影是视频数据。我们现在常见的网页包含文字、图片、声音、视频等数据。这里所说的数据必须具备两个前提:

- 可以输入到计算机中。
- 能被计算机程序处理。

无论是文字、图片、声音还是视频数据, 在计算机内部, 最终都以二进制形式表示。

2. 数据元素

数据元素(Data Element)是组成数据的有一定意义的基本单位, 在计算机中通常作为整体考虑和处理。例如, 一个数据元素可以由若干个数据项组成, 数据项是数据不可分割的最小单位。学生信息表包括学号、姓名、性别、籍贯、所在院系、出生日期、联系电话等数据项。这里的数据元素也称为记录。学生信息表如表 1.1 所示。

表 1.1 学生信息表

学号	姓名	性别	籍贯	所在院系	出生日期	联系电话
201208001	赵倩	女	陕西	信息学院	1994.12	88306200
201203002	吴江	男	河南	化学系	1993.08	88306512
201201003	王平静	男	陕西	文学院	1994.11	88308256

比如, 如果将人看作是数据元素的话, 那么口、眼、鼻、嘴、耳、手就是数据项, 姓名、年龄、性别、出生日期、联系电话等也可以看作数据项。

3. 数据对象

数据对象(Data Object)是性质相同的数据元素的集合, 是数据的子集。例如, 正整数的数据对象是集合 $N=\{1, 2, 3, \dots\}$, 字母字符的数据对象是集合 $C=\{'A', 'B', 'C', \dots, 'Z'\}$ 。

4. 数据结构

结构, 简单地理解就是关系, 比如分子结构就是组成分子的原子之间的排列方式。在现实世界中, 数据元素并不是孤立存在的, 而是具有某种内在联系的, 我们将这些关系称为结构。数据结构(Data Structure)是数据元素之间存在的一种或多种特定关系, 也就是数据的组织形式。在计算机中, 数据元素的关系也不是孤立的、无序的, 而是具有内在联系的

数据集合。计算机中的数据元素之间的关系是对现实世界的抽象，数据结构这门课将要详细阐述3种结构：表结构、层次结构和图结构。

例如，表1.1的学生信息表是一种表结构，如图1.1所示的学校组织结构图是一种层次结构，如图1.2所示的城市之间的交通路线图是一种图结构。

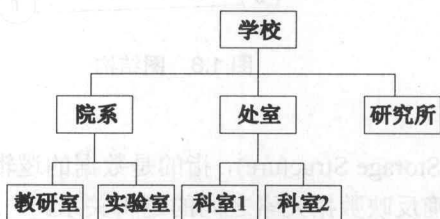


图 1.1 学校的组织结构图

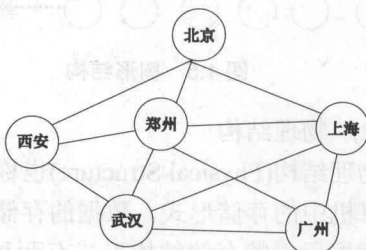


图 1.2 城市之间的交通路线图

5. 数据结构的逻辑结构与存储结构

数据结构的主要任务就是通过分析要描述对象的结构特征，包括逻辑结构及内在联系，也就是抽象数据类型中所要描述的数据关系，把逻辑结构表示成计算机可实现的物理结构，从而方便计算机处理。

(1) 逻辑结构

数据的逻辑结构是指在数据对象中，数据元素之间的相互关系。数据元素之间在不同的逻辑关系，构成了以下4种结构类型。

① 集合。结构中的数据元素除了同属于一个集合外，数据元素之间没有其他关系。这就像我们数学中的自然数集合，集合中的所有元素都属于该集合，除此之外，没有其他特性。例如，数学中的正整数集合 $\{12, 78, 56, 52, 20, 90\}$ ，集合中的数除了属于正整数外，元素之间没有其他关系，数据结构中的集合关系就类似于数学中的集合。集合表示如图1.3所示。

② 线性结构。结构中的数据元素之间是一一对应的关系。线性结构如图1.4所示。数据元素之间有一种先后的次序关系， a, b, c, d, e, f 构成一个线性表，其中， a 是 b 的前驱， b 是 a 的后继。

③ 树形结构。结构中的数据元素之间存在一种一对多的层次关系，树形结构如图1.5所示。这就像学校的组织结构图，学校下面是教学的院系、行政机构的部和处及一些研究所。

④ 图结构。结构中的数据元素是多对多的关系。图结构如图1.6所示。城市之间的交通路线图就是多对多的关系， A, B, C, D, E, F 是六个城市，城市 A 和城市 B, C, D 都存在一条直达路线，而城市 B 也和 A, D, F 存在一条直达路线，从图1.6中可以看出城市之间的关系是错综复杂的。

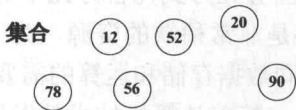


图 1.3 集合结构

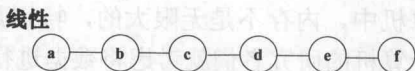


图 1.4 线性结构

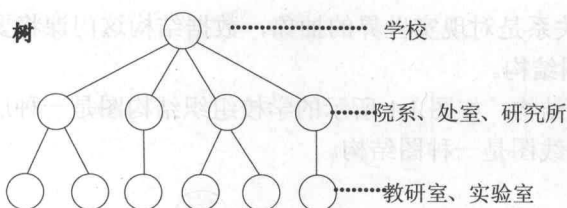


图 1.5 树形结构

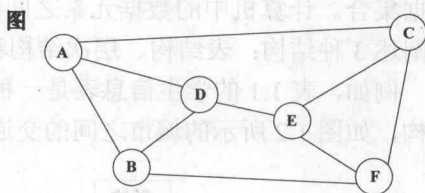


图 1.6 图结构

(2) 物理结构

物理结构(Physical Structure)也称为存储结构(Storage Structure),指的是数据的逻辑结构在计算机中的存储形式。数据的存储结构应能正确反映数据元素之间的逻辑关系。

数据元素的存储结构形式有两种:顺序存储结构和链式存储结构。顺序存储是把数据元素存放在一块地址连续的存储单元里,其数据间的逻辑关系和物理关系是一致的。顺序存储结构如图 1.7 所示。

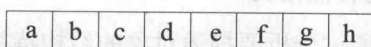


图 1.7 顺序存储结构

链式存储是把数据元素存放在任意的存储单元里,这组存储单元可以是连续的,也可以是不连续的,数据元素的存储关系并不能反映其逻辑关系,因此需要用一个指针存放数据元素的地址,通过地址就可以找到相关联数据元素的位置。链式存储结构如图 1.8 所示。

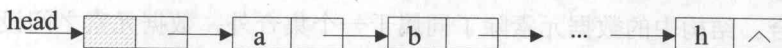


图 1.8 链式存储结构

数据的逻辑结构和物理结构是数据对象的逻辑表示和物理表示,数据结构要对建立起来的逻辑结构和物理结构进行处理,就需要建立起计算机可以运行的程序集合。

6. 数据类型

数据类型(Data Type)是用来刻画一组性质相同的数据及其上的操作的总称。数据类型是按照值的不同进行划分的。在高级语言中,每个变量、常量和表达式都有各自的取值范围,该类型就说明了变量或表达式的取值范围和所能进行的操作。例如,C语言中的字符类型所占空间是 8 位,这就是它的取值范围,在其范围内可以进行赋值运算、比较运算等。

也许读者不禁要问,当年那些设计计算机语言的牛人们为什么会用到数据类型呢?这其实就像我们买汽车,都希望功能全、排量大、性能好,但是没有钱的平民百姓考虑太贵的车不现实,为了满足各种用户需求,于是就出现了各种各样的汽车,自主品牌的相对来说便宜,合资的要贵一些,排量从 0.8L 到 6.0L,价位也从四五万元到几百万元不等。同样,在计算机中,内存不是无限大的,特别是几十年前,内存是非常稀缺的资源,为了节省资源,计算机的研究者们就考虑将数据进行分类,以满足不同数据存储和运算的需要。

在计算机刚刚诞生时,只能处理一些简单的数值信息,随着计算机技术的发展,计算机所能处理的对象已经囊括数值、字符、文字、声音、图像、视频等信息。任何信息只要

经过数字化处理,都能够让计算机识别。当然,这需要对处理的信息进行抽象描述,让计算机理解。

在C语言中,按照取值的不同,数据类型还可以分为两类——原子类型和结构类型。

- 原子类型:即不可以再分解的基本类型,包括整型、实型、字符型等。
- 结构类型:是由若干个类型组合而成,可以再分解的。例如,整型数组是由若干整型数据组成的,结构体类型的值也是由若干个类型范围的数据构成,它们的类型都是相同的。

例如,在C语言中,如果有变量声明 `int a, b;`,则变量 `a` 和 `b` 在被赋值时不能超出 `int` 的取值范围,变量 `a` 和 `b` 的运算只能是 `int` 类型所允许的运算。

1.2 抽象数据类型

在计算机处理过程中,需要把处理的对象抽象出来,描述成计算机能理解的形式,也就是把数据信息符号化,表示成一个模型,而有时仅仅用基本的数据类型是不够的,这就需要抽象数据类型的描述。

1.2.1 抽象数据类型的定义

抽象数据类型(Abstract Data Type, ADT)是对具有某种逻辑关系的数据类型进行描述,并在该类型上进行的一组操作。抽象数据类型描述的是一组逻辑上的特性,与在计算机内部如何表示及如何实现无关。计算机中的整数数据类型是一个抽象数据类型,不同的处理器可能实现方法不同,但其逻辑特性相同,即加、减、乘、除等运算是是一致的。

抽象数据类型不仅包括在计算机中已经定义了的数据类型,例如整型、浮点型等,还包括用户自己定义的数据类型,例如结构体类型、类等。

一个抽象数据类型定义了一个数据对象、数据元素之间的关系及对数据元素的操作。抽象数据类型通常是指用户定义的解决应用问题的数据模型,包括数据的定义和操作。例如,C++的类就是一个抽象数据类型,它包括用户类型的定义和在用户类型上的一组操作。

抽象数据类型体现了程序设计中的问题分解、抽象和信息隐藏特性。抽象数据类型把实际生活中的问题分解为多个规模小且容易处理的问题,然后建立起一个计算机能处理的数据模型,并把每个功能模块的实现细节作为一个独立的单元,从而将具体的实现过程隐藏起来。

这就类似于日常生活中盖房子,我们可以把盖房子分成几个小任务,一方面需要工程技术人员提供房子的设计图纸,另一方面需要建筑工人根据图纸打地基、盖房子,房子盖好以后,还需要装修工人装修,这与抽象数据类型中的问题分解类似。工程技术人员不需要了解打地基和盖房子的具体过程,装修工人也不需要知道怎么画图纸和怎样盖房子。

抽象数据类型中的信息隐藏也是如此,每一个基本操作不需要了解其他基本操作的实现过程。



1.2.2 抽象数据类型的描述

抽象数据类型常见的描述方式如下：

ADT 抽象数据类型名

{

数据对象：<数据对象的定义>

数据关系：<数据关系的定义>

基本操作：<基本操作的定义>

} ADT 抽象数据类型名

为了简便和容易理解，本书用以下方式描述抽象数据类型。

抽象数据类型可分为数据集合和基本操作集合。其中，数据集合包括对数据对象和数据对象中元素之间关系的描述，基本操作集合是对数据对象的运算的描述。数据对象和数据关系的定义采用数学符号和自然语言描述，基本操作的定义格式为：

基本操作名(参数表)：初始条件和操作结果描述。

例如，一个队列的抽象数据类型描述如下。

1. 数据集合

队列的数据集合为 $\{a_1, a_2, \dots, a_n\}$ ，每个元素的类型均为 **DataType**。其中， a_1 是队头元素， a_n 是队尾元素。入队和出队都是按照 $a_1, a_2, \dots, a_n$ 的先后次序进入队列和退出队列。

2. 基本操作集合

队列的基本操作主要有以下几种。

(1) **InitQueue(&Q)**：队列的初始化操作。这就像日常生活中，火车站售票处新增加了一个售票窗口，这样就可以新增一队用来排队买票。

初始条件：队列 Q 不存在。

操作结果：构造一个空队列 Q 。

(2) **QueueEmpty(Q)**：判断队列是否为空。这就像日常生活中，火车站售票窗口前是否还有人排队买票。

初始条件：队列 Q 已存在。

操作结果：若 Q 为空队列，返回 1；否则返回 0。

(3) **EnQueue(&Q, e)**：队列的插入操作。这就像日常生活中，新来买票的人要在队列的最后一样。

初始条件：队列 Q 已存在。

操作结果：插入元素 e 到队列 Q 的队尾。

(4) **DeQueue(&Q, &e)**：队列的删除操作。这就像买过票的排在队头的人离开队列。

初始条件：队列 Q 已存在。

操作结果：删除 Q 的队头元素，并用 e 返回其值。

(5) **Gethead(Q, &e)**：取队头元素操作。这就像询问排队买票的人是谁。

初始条件：队列 Q 已存在且非空。