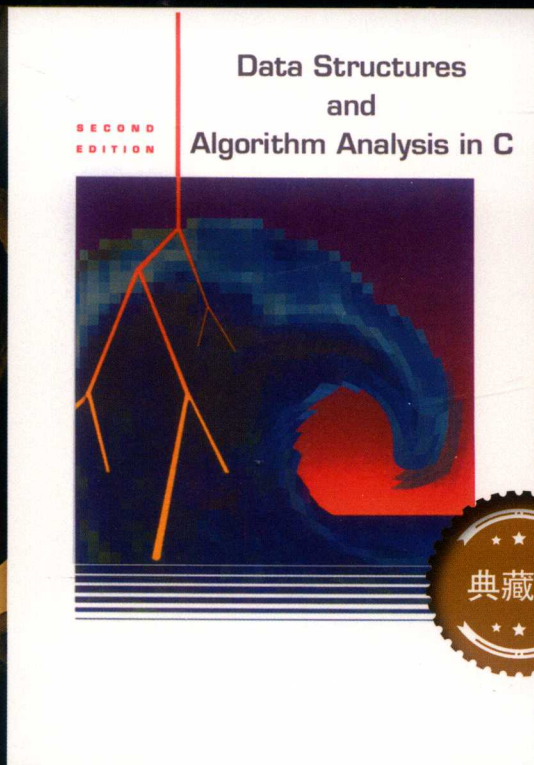


# 数据结构与算法分析

## C语言描述

[美] 马克·艾伦·维斯 (Mark Allen Weiss) 著  
冯舜玺 译

Data Structures and Algorithm Analysis in C  
Second Edition



计

丛

书

原书第2版

# 数据结构与算法分析

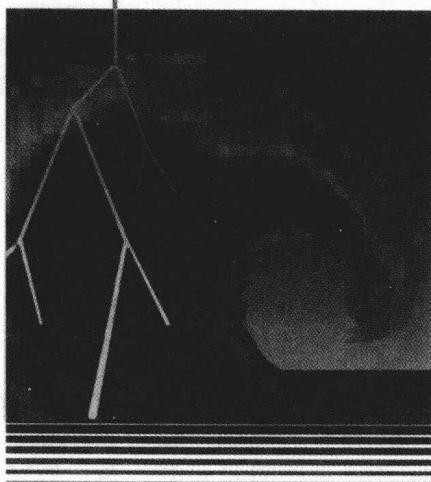
## C语言描述

[美] 马克·艾伦·维斯 (Mark Allen Weiss) 著  
冯舜玺 译

Data Structures and Algorithm Analysis in C  
Second Edition

SECOND  
EDITION

Data Structures  
and  
Algorithm Analysis in C



RFLD



典藏版



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

数据结构与算法分析——C 语言描述 (原书第 2 版) (典藏版) / (美) 马克·艾伦·维斯 (Mark Allen Weiss) 著; 冯舜玺译. —北京: 机械工业出版社, 2019.3  
(计算机科学丛书)

书名原文: Data Structures and Algorithm Analysis in C, Second Edition

ISBN 978-7-111-62195-9

I. 数… II. ①马… ②冯… III. ①数据结构—研究生—教材 ②算法分析—研究生—教材  
③ C 语言—程序设计—研究生—教材 IV. ① TP311.12 ② TP312.8

中国版本图书馆 CIP 数据核字 (2019) 第 043641 号

本书版权登记号: 图字 01-2018-8479

Authorized translation from the English language edition, entitled *Data Structures and Algorithm Analysis in C, Second Edition*, 9780201498400 by Mark Allen Weiss, published by Pearson Education, Inc., Copyright © 1997.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2019.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

本书是国外数据结构与算法分析方面的经典教材, 介绍了数据结构 (大量数据的组织方法) 以及算法分析 (算法运行时间的估算)。本书的编写目标是同时讲授好的程序设计和算法分析技巧, 使读者可以开发出高效的程序。

本书可作为高级数据结构课程或研究生一年级算法分析课程的教材, 使用本书需要一些中级程序设计知识, 还需要离散数学的一些背景知识。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 唐晓琳

责任校对: 李秋荣

印刷: 北京诚信伟业印刷有限公司

版次: 2019 年 4 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 25.75

书号: ISBN 978-7-111-62195-9

定价: 79.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88379833

投稿热线: (010) 88379604

购书热线: (010) 68326294

读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson、McGraw-Hill、Elsevier、MIT、John Wiley & Sons、Cengage等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Andrew S. Tanenbaum、Bjarne Stroustrup、Brian W. Kernighan、Dennis Ritchie、Jim Gray、Afred V. Aho、John E. Hopcroft、Jeffrey D. Ullman、Abraham Silberschatz、William Stallings、Donald E. Knuth、John L. Hennessy、Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近500个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：[www.hzbook.com](http://www.hzbook.com)

电子邮件：[hzjsj@hzbook.com](mailto:hzjsj@hzbook.com)

联系电话：(010)88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



## 译者序 T H E T R A N S L A T O R ' S W O R D S

随着速度的不断提高和存储容量的持续增长，计算机的功能日益强大，从而处理数据和解决问题的规模和复杂程度与日俱增。这不仅带来了需要认真研究的新课题，而且突出了原有数据结构和算法效率低下的缺点。程序的效率问题并没有由于计算机功能的强大而受到冷落，相反，被人们提到前所未有的重要地位，因为大型问题的解决所涉及的大容量存储和高速度运算容不得我们对效率有丝毫的忽视。本书正是在阐述数据结构基本概念的同时深入地分析了算法的效率。书中详细介绍了当前流行的论题和新的变化，讨论了算法设计技巧，并在研究算法的性能、效率以及分析运行时间的基础上考察了一些高级数据结构，从历史的角度和近年的进展对数据结构的活跃领域进行了简要的概括。由于本书原版选材新颖，方法实用，题例丰富，取舍得当，因此，自出版以来受到广泛欢迎，已被世界许多知名大学用作教材。

本书的目的是培养学生良好的程序设计技巧和熟练的算法分析能力，使得他们能够开发出高效率的程序。从服务于实践和锻炼学生实际能力出发，书中提供了大部分算法的 C 语言和伪代码例程，一些程序可从互联网上获得。

承蒙卢开澄教授、温莉芳女士的鼓励，译者有幸将国外几部优秀原著介绍给我国的读者，在此表示衷心的感谢。译者还想借此机会感谢挚友孙华先生，他对本书的翻译工作自始至终给予热心的关怀和无私的帮助。

由于时间及水平所限，书中译文不当之处，统祈学术界同仁及广大读者赐正。

译者

## 目的

本书讨论数据结构和算法分析。数据结构主要研究组织大量数据的方法，而算法分析则是对算法运行时间的评估。随着计算机的速度越来越快，对于能够处理大量输入数据的程序的需求变得日益急切。可是，由于在输入量很大的时候程序的低效率现象变得非常明显，因此这又要求对效率问题给予更仔细的关注。通过在实际编程前对算法进行分析，学生可以决定一个特定的解法是否可行。例如，学生在本书中将读到一些特定的问题并看到精心的实现方法是如何把处理大量数据的时间限制从 16 年减至不到 1 秒的。因此，若无运行时间的阐释，就不会有算法和数据结构的提出。在某些情况下，对于影响算法实现的运行时间的一些微小细节都需要认真探究。

一旦确定解法，还必须编写程序。随着计算机的日益强大，它们必须解决的问题也变得更加巨大和复杂，这就要求开发更加复杂的程序。本书的目的是教授学生良好的程序设计技巧和提高学生的算法分析能力，使得他们能够开发出具有最高效率的程序。

本书适合作为高级数据结构(CS7)课程或研究生第一年算法分析课程的教材。学生应该具有中等程度的程序设计知识，包括像指针和递归这样一些内容，还应该具有离散数学的某些知识。

## 方法

我相信，对于学生来说，重要的是学习如何自己动手编写程序，而不是从书上拷贝程序。但另一方面，讨论现实程序设计问题而不套用样本程序实际上是不可能的。由于这个原因，本书通常提供实现方法的大约一半到四分之三的内容并鼓励学生补足其余的部分。第 12 章是这一版新加的，讨论主要侧重于实现细节的一些附加的数据结构。

本书中的算法均以 ANSI C 表示，尽管有些欠缺，但它仍然是最流行的系统程序设计语言。使用 C 代替 Pascal，使得动态分配数组成为可能(见第 5 章中的“再散列”)。它还在几处地方将代码简化，这通常是与(&&)操作走捷径的缘故。

对 C 的大多数批评集中在用它写出的程序代码可读性差的事实上。仅仅少击几次键，却牺牲了程序的清晰性，而程序的速度又没有增加。因此，诸如同时赋值以及通过

```
if(x=y)
```

测试是否为 0 等技巧一般不在本书中使用。本书将证明只要细心练习是可以避免那些难以读懂的代码的。

## 内容提要

第 1 章包含离散数学和递归的一些复习材料。我相信对递归做到泰然处之的唯一办法是反复不断地看一些好的用法。因此，除第 5 章外，递归遍及本书每一章的例子之中。

第 2 章处理算法分析。该章阐述渐近分析和它的主要弱点。这里提供了许多例子，包括对对数运行时间的深入解释。通过直观地把一些简单递归程序转变成迭代程序而对它们进行分析。介绍了更为复杂的分治程序，不过有些分析(求解递归关系)要推迟到第 7 章再详细讨论。

第 3 章包括表、栈和队列。重点聚焦于使用 ADT 对这些数据结构编程，这些数据结构的快速实现，以及介绍它们的某些用途。文中几乎没有什么程序(只有些例程)，而程序设计作业的许多思想基本上体现在练习之中。

第 4 章讨论树，重点在于查找树，包括外部查找树(B 树)。UNIX 文件系统和表达式树是作为例子来介绍的。AVL 树和伸展树只进行了介绍而没有分析。程序写出 75%，其余部分留给学生完成。查找树的实现细节见第 12 章。树的另外一些内容，如文件压缩和博弈树，延迟到第 10 章讨论。外部媒体上的数据结构在这几章的最后讨论。

第 5 章是相对较短的一章，主要讨论散列表。这里进行了某些分析，该章末尾讨论了可扩散列。

第 6 章讨论优先队列。二叉堆也在该章讲授，还有些附加的材料论述优先队列某些理论上有趣的实现方法。斐波那契堆在第 11 章讨论，配对堆在第 12 章讨论。

第 7 章讨论排序。它特别关注编程细节和分析，讨论并比较所有通用的排序算法。对以下四种算法进行了详细的分析：插入排序、希尔排序、堆排序以及快速排序。堆排序平均情形运行时间的分析对于这一版来说是新的内容。该章末尾讨论了外部排序。

第 8 章讨论不相交集算法并证明其运行时间。该章短而专，如果不讨论 Kruskal 算法则可跳过。

第 9 章讲授图论算法。图论算法很重要，不仅因为在实践中经常用到它们，而且还因为它们的运行时间强烈地依赖于数据结构的恰当使用。实际上，所有标准算法都是和相应的数据结构、伪代码以及运行时间的分析一起介绍的。为把这些问题放进一本适当的教材中，我们对复杂性理论(包括 NP-完全性和不可判定性)进行了简短的讨论。

第 10 章通过考察一般的问题求解技巧讨论算法设计。该章添加了大量的实例。这里及后面各章使用的伪代码使得学生能更好地理解例子，从而避免被实现的细节干扰。

第 11 章处理摊还分析。对来自第 4 章到第 6 章的三种数据结构以及该章介绍的斐波那契堆进行了分析。

第 12 章是这一版新加的，讨论查找树算法、 $k$  维( $k$ -d)树和配对堆。不同于其他各章，该章给出了查找树和配对堆完整详细的实现。教师可以把一些内容纳入其他各章的讨论之中。例如，第 12 章中的自顶向下红黑树可以在第 4 章的 AVL 树下讨论。

第 1 章到第 9 章为大多数的一学期数据结构课程提供了足够的材料。如果时间允许，那么第 10 章也可以包括进来。研究生的算法分析课程可以使用第 7 章到第 11 章的内容。第 11 章所分析的高级数据结构可以容易地在前面各章中查到。第 9 章中对 NP-完全性的讨论对于这门课来说太过简要，Garey 和 Johnson 的论 NP-完全性的书可以补充本书的不足。

## 练习

每章末尾提供的练习与书中讲授的内容顺序相匹配。最后的一些练习针对整个一章而不是特定的某一节。难做的练习以一个星号标记，更难做的练习标有两个星号。

教师可从 Addison-Wesley 出版公司得到包含几乎所有练习答案的解题指南<sup>⊖</sup>。

## 参考文献

参考文献位于每章的最后。一般说来，这些参考文献或者是历史性的，代表着书中材料的原始来源，或者阐述对书中给出的结果的扩展和改进。有些文献论述了一些练习的解法。

## 代码的获得

本书中的程序代码可通过匿名 ftp 在 aw.com 网站得到。这个网站也可以通过 World Wide Web 来访问，其 URL 为 <http://www.aw.com/cseng/>(从此处继续链接)。该资料的

---

⊖ 关于本书教辅资源，只有使用本书作为教材的教师才可以申请，需要的教师请联系机械工业出版社华章公司，电话 010-88378991，邮箱 wangguang@hzbook.com。——编辑注



准确位置可能变化。

## 致谢

在几部著作的准备过程中，本人得到许多朋友的帮助。有些人在本书的其他版本中提到过，谢谢诸位。

对于这一版，我要感谢 Addison-Wesley 的编辑 Carter Shanklin 和 Susan Hartman。Teri Hyde 完善了本书的出版工作，而 Matthew Harris 和他在出版服务中心的同事出色地完成了本书最后的定稿任务。

M. A. W.

Miami, Florida

1996 年 7 月

出版者的话

译者序

前言

## 第 1 章 引论 | 1

1.1 本书讨论的内容 | 2

1.2 数学知识复习 | 3

1.2.1 指数 | 3

1.2.2 对数 | 3

1.2.3 级数 | 4

1.2.4 模运算 | 5

1.2.5 证明方法 | 5

1.3 递归简论 | 7

总结 | 10

练习 | 10

参考文献 | 11

## 第 2 章 算法分析 | 13

2.1 数学基础 | 14

2.2 模型 | 16

2.3 要分析的问题 | 16

2.4 运行时间计算 | 18

2.4.1 一个简单的例子 | 18

2.4.2 一般法则 | 19

2.4.3 最大子序列和 | 20

2.4.4 运行时间中的对数 | 24

2.4.5 检验你的分析 | 27

2.4.6 分析结果的准确性 | 28

总结 | 28

练习 | 29

参考文献 | 32

## 第 3 章 表、栈和队列 | 35

3.1 抽象数据类型 | 36

3.2 表 ADT | 36

3.2.1 表的简单数组实现 | 37

3.2.2 链表 | 37

3.2.3 程序设计细节 | 38

3.2.4 常见的错误 | 42

3.2.5 双链表 | 43

3.2.6 循环链表 | 43

3.2.7 例子 | 43

3.2.8 链表的游标实现 | 47

3.3 栈 ADT | 50

3.3.1 栈模型 | 50

3.3.2 栈的实现 | 51

3.3.3 应用 | 56

3.4 队列 ADT | 62

3.4.1 队列模型 | 62

3.4.2 队列的数组实现 | 62

3.4.3 队列的应用 | 65

总结 | 66

练习 | 66

**第4章 树** | 71

- 4.1 预备知识 | 72
  - 4.1.1 树的实现 | 73
  - 4.1.2 树的遍历及应用 | 74
- 4.2 二叉树 | 76
  - 4.2.1 实现 | 77
  - 4.2.2 表达式树 | 77
- 4.3 查找树 ADT——二叉查找树 | 80
  - 4.3.1 MakeEmpty | 80
  - 4.3.2 Find | 81
  - 4.3.3 FindMin 和 FindMax | 81
  - 4.3.4 Insert | 81
  - 4.3.5 Delete | 83
  - 4.3.6 平均情形分析 | 84
- 4.4 AVL 树 | 86
  - 4.4.1 单旋转 | 88
  - 4.4.2 双旋转 | 90
- 4.5 伸展树 | 95
  - 4.5.1 一个简单的想法 | 96
  - 4.5.2 展开 | 97
- 4.6 树的遍历 | 102
- 4.7 B 树 | 103
- 总结 | 107
- 练习 | 108
- 参考文献 | 113

**第5章 散列** | 117

- 5.1 一般想法 | 118
- 5.2 散列函数 | 118
- 5.3 分离链接法 | 120
- 5.4 开放定址法 | 123
  - 5.4.1 线性探测法 | 124
  - 5.4.2 平方探测法 | 125
  - 5.4.3 双散列 | 129
- 5.5 再散列 | 130
- 5.6 可扩散列 | 132
- 总结 | 133

- 练习 | 134
- 参考文献 | 137

**第6章 优先队列(堆)** | 139

- 6.1 模型 | 140
- 6.2 一些简单的实现 | 141
- 6.3 二叉堆 | 141
  - 6.3.1 结构性质 | 141
  - 6.3.2 堆序性质 | 142
  - 6.3.3 基本的堆操作 | 143
  - 6.3.4 其他的堆操作 | 146
- 6.4 优先队列的应用 | 149
  - 6.4.1 选择问题 | 149
  - 6.4.2 事件模拟 | 150
- 6.5 d-堆 | 151
- 6.6 左式堆 | 152
  - 6.6.1 左式堆的性质 | 152
  - 6.6.2 左式堆的操作 | 153
- 6.7 斜堆 | 158
- 6.8 二项队列 | 159
  - 6.8.1 二项队列结构 | 159
  - 6.8.2 二项队列操作 | 160
  - 6.8.3 二项队列的实现 | 162
- 总结 | 165
- 练习 | 166
- 参考文献 | 169

**第7章 排序** | 173

- 7.1 预备知识 | 174
- 7.2 插入排序 | 174
  - 7.2.1 算法 | 174
  - 7.2.2 插入排序的分析 | 175
- 7.3 一些简单排序算法的下界 | 175
- 7.4 希尔排序 | 176
- 7.5 堆排序 | 179
- 7.6 归并排序 | 182
- 7.7 快速排序 | 186

- 7.7.1 选取枢纽元 | 187
- 7.7.2 分割策略 | 188
- 7.7.3 小数组 | 190
- 7.7.4 实际的快速排序例题 | 190
- 7.7.5 快速排序的分析 | 192
- 7.7.6 选择的线性期望时间算法 | 194
- 7.8 大型结构的排序 | 195
- 7.9 排序的一般下界 | 196
- 7.10 桶式排序 | 198
- 7.11 外部排序 | 198
  - 7.11.1 为什么需要新的算法 | 198
  - 7.11.2 外部排序模型 | 199
  - 7.11.3 简单算法 | 199
  - 7.11.4 多路合并 | 200
  - 7.11.5 多相合并 | 201
  - 7.11.6 替换选择 | 202

总结 | 203

练习 | 204

参考文献 | 207

## 第 8 章 不相交集 ADT | 209

- 8.1 等价关系 | 210
- 8.2 动态等价性问题 | 210
- 8.3 基本数据结构 | 212
- 8.4 灵巧求并算法 | 214
- 8.5 路径压缩 | 216
- 8.6 按秩求并和路径压缩的最坏情形 | 217
- 8.7 一个应用 | 221

总结 | 222

练习 | 222

参考文献 | 223

## 第 9 章 图论算法 | 225

- 9.1 若干定义 | 226
- 9.2 拓扑排序 | 228
- 9.3 最短路径算法 | 230
  - 9.3.1 无权最短路径 | 232

- 9.3.2 Dijkstra 算法 | 235
- 9.3.3 具有负边值的图 | 240
- 9.3.4 无圈图 | 241
- 9.3.5 所有点对最短路径 | 243

9.4 网络流问题 | 243

9.5 最小生成树 | 247

9.5.1 Prim 算法 | 248

9.5.2 Kruskal 算法 | 250

9.6 深度优先搜索的应用 | 251

9.6.1 无向图 | 252

9.6.2 双连通性 | 253

9.6.3 欧拉回路 | 256

9.6.4 有向图 | 259

9.6.5 查找强分支 | 260

9.7 NP-完全性介绍 | 262

9.7.1 难与易 | 262

9.7.2 NP 类 | 263

9.7.3 NP-完全问题 | 264

总结 | 266

练习 | 266

参考文献 | 270

## 第 10 章 算法设计技巧 | 273

10.1 贪婪算法 | 274

10.1.1 一个简单的调度问题 | 274

10.1.2 Huffman 编码 | 276

10.1.3 近似装箱问题 | 280

10.2 分治算法 | 286

10.2.1 分治算法的运行时间 | 287

10.2.2 最近点问题 | 289

10.2.3 选择问题 | 291

10.2.4 一些运算问题的理论改进 | 294

10.3 动态规划 | 297

10.3.1 用一个表代替递归 | 298

10.3.2 矩阵乘法的顺序安排 | 300

10.3.3 最优二叉查找树 | 301

10.3.4 所有点对最短路径 | 304

- 10.4 随机化算法 | 306
  - 10.4.1 随机数发生器 | 307
  - 10.4.2 跳跃表 | 310
  - 10.4.3 素性测试 | 312
- 10.5 回溯算法 | 314
  - 10.5.1 收费公路重建问题 | 314
  - 10.5.2 博弈 | 318

总结 | 323

练习 | 323

参考文献 | 329

## 第 11 章 摊还分析 | 333

- 11.1 一个无关的智力问题 | 334
- 11.2 二项队列 | 335
- 11.3 斜堆 | 339
- 11.4 斐波那契堆 | 341
  - 11.4.1 切除左式堆中的节点 | 341
  - 11.4.2 二项队列的懒惰合并 | 343
  - 11.4.3 斐波那契堆操作 | 346
  - 11.4.4 时间界的证明 | 346

11.5 伸展树 | 348

总结 | 351

练习 | 351

参考文献 | 353

## 第 12 章 高级数据结构及其实现 | 355

12.1 自顶向下伸展树 | 356

12.2 红黑树 | 361

12.2.1 自底向上插入 | 362

12.2.2 自顶向下红黑树 | 363

12.2.3 自顶向下删除 | 367

12.3 确定性跳跃表 | 368

12.4 AA 树 | 373

12.5 treap 树 | 378

12.6 k-d 树 | 379

12.7 配对堆 | 383

总结 | 387

练习 | 387

参考文献 | 389

索引 | 391

# 第 1 章

## 引 论

在这一章，我们阐述本书的目的，并简要复习离散数学以及程序设计的一些概念。我们将：

- 看到程序在较大输入情况下的运行性能与在适量输入情况下的运行性能具有同等重要性。
- 总结本书其余部分所需要的数学基础。
- 简要复习递归。

## 1.1 本书讨论的内容

设有一组  $N$  个数而要确定其中第  $k$  个最大者。我们称之为选择问题 (selection problem)。大多数学习过一两门程序设计课程的学生写一个解决这种问题的程序不会有什么困难。“显而易见的”解决方法有很多。

该问题的一种解法就是将这  $N$  个数读进一个数组中，再通过某种简单的算法，比如冒泡排序法，以递减顺序将数组排序，然后返回位置  $k$  上的元素。

稍微好一点的算法可以先把前  $k$  个元素读入数组并(以递减的顺序)对其排序。接着，将剩下的元素再逐个读入。当读取新元素时，如果它小于数组中的第  $k$  个元素则忽略，否则就将其放到数组中正确的位置上，同时将数组中的一个元素挤出数组。当算法终止时，位于第  $k$  个位置上的元素作为答案返回。

这两种算法的编码都很简单，建议读者试一试。此时我们自然要问：哪个算法更好？哪个算法更重要？还是两个算法都足够好？使用含有 100 万个元素的随机文件，在  $k=500\,000$  的条件下进行模拟发现，两个算法在合理的时间内均不能结束，每种算法都需要计算机处理若干天才能算完(虽然最后还是给出了正确的答案)。在第 7 章将讨论另一种算法，该算法将在 1 秒左右给出问题的解。因此，虽然我们提出的两个算法都能算出结果，但是不能认为它们是好的算法，因为对于第三种算法在合理的时间内能够处理的输入数据量而言，这两种算法是完全不切实际的。

第二个问题是解决一个流行的字谜。输入由一些含字母的二维数组和一个单词列表组成。目标是要找出字谜中的单词，这些单词可能是水平、垂直或沿对角线以任何方向放置的。作为例子，图 1-1 所示的字谜由单词 this、two、fat 和 that 组成。单词 this 从第一行第一列的位置即(1, 1)处开始并延伸至(1, 4)；单词 two 从(1, 1)到(3, 1)；fat 从(4, 1)到(2, 3)；而 that 则从(4, 4)到(1, 1)。

	1	2	3	4
1	t	h	i	s
2	w	a	t	s
3	o	a	h	g
4	f	g	d	t

图 1-1 字谜示例

现在至少有两种直观的算法来求解这个问题。对单词表中的每个单词，我们检查每一个有序三元组(行，列，方向)，验证是否有单词存在。这需要大量嵌套的 for 循环，但它基本上是直观的算法。

或者，对于每一个尚未进行到字谜最后的有序四元组(行，列，方向，字符数)我们可以测试所指的单词是否在单词表中。这也导致使用大量嵌套的 for 循环。如果在任意单词

中的最大字符数已知，那么该算法有可能节省一些时间。

上述两种方法相对来说都不难编码，并可求解通常发表于杂志上的许多现实的字谜游戏。这些字谜通常有 16 行 16 列以及 40 个左右的单词。然而，假设我们把字谜变为只给出谜板(puzzle board)而单词表基本上是一本英语词典，则上面提出的两种解法需要相当可观的时间来解决这个问题，故这两种方法都是不可接受的。不过，这样的问题还是有可能在数秒内解决的，即使单词表很大也可以。

在许多问题当中，一个重要的观念是：写出一个可以工作的程序并不够。如果这个程序在巨大的数据集上运行，那么运行时间就变成了重要的问题。我们将在本书中看到对于大量的输入如何估计程序的运行时间，尤其是如何在尚未具体编码的情况下比较两个程序的运行时间。我们还将看到彻底改进程序速度以及确定程序瓶颈的方法。这些方法将使我们能够找到需要大力优化的那些代码段。

2

## 1.2 数学知识复习

本节列出一些需要记住或是能够推导出的基本公式，复习基本的证明方法。

### 1.2.1 指数

$$\begin{aligned} X^A X^B &= X^{A+B} \\ \frac{X^A}{X^B} &= X^{A-B} \\ (X^A)^B &= X^{AB} \\ X^N + X^N &= 2X^N \neq X^{2N} \\ 2^N + 2^N &= 2^{N+1} \end{aligned}$$

### 1.2.2 对数

在计算机科学中，除非有特别的声明，所有的对数都是以 2 为底的。

**定义：**当且仅当  $\log_x B = A$ ,  $X^A = B$ 。

由该定义可以得到几个方便的等式。

**定理 1.1**

$$\log_A B = \frac{\log_C B}{\log_C A}; C > 0$$

**证明：**令  $X = \log_C B$ ,  $Y = \log_C A$ , 以及  $Z = \log_A B$ 。此时由对数的定义得： $C^X = B$ ,  $C^Y = A$  以及  $A^Z = B$ 。联合这三个等式则产生  $(C^Y)^Z = C^X = B$ 。因此， $X = YZ$ ，这意味着  $Z = X/Y$ ，定理得证。

**定理 1.2**

$$\log AB = \log A + \log B$$

**证明：**令  $X = \log A$ ,  $Y = \log B$ , 以及  $Z = \log AB$ 。此时由于假设默认的底为 2,  $2^X = A$ ,  $2^Y = B$  及  $2^Z = AB$ 。联合最后的三个等式则有  $2^X 2^Y = 2^Z = AB$ 。因此  $X + Y = Z$ ，这就证



明了该定理。

其他一些有用的公式如下，它们都能够用类似的方法推导。

$$\log A/B = \log A - \log B$$

$$\log(A^B) = B \log A$$

$\log X < X$  (对所有的  $X > 0$  成立)

$$\log 1 = 0, \log 2 = 1, \log 1024 = 10, \log 1\,048\,576 = 20$$

### 1.2.3 级数

最容易记忆的公式是

$$\sum_{i=0}^N 2^i = 2^{N+1} - 1$$

和

$$\sum_{i=0}^N A^i = \frac{A^{N+1} - 1}{A - 1}$$

在第二个公式中，如果  $0 < A < 1$ ，则

$$\sum_{i=0}^N A^i \leq \frac{1}{1-A}$$

当  $N$  趋于  $\infty$  时该和趋向于  $1/(1-A)$ ，这些公式是“几何级数”公式。

可以用下面的方法推导关于  $\sum_{i=0}^{\infty} A^i$  ( $0 < A < 1$ ) 的公式。令  $S$  表示和，此时

$$S = 1 + A + A^2 + A^3 + A^4 + A^5 + \dots$$

于是

$$AS = A + A^2 + A^3 + A^4 + A^5 + \dots$$

如果将这两个等式相减(这种运算只能对收敛级数进行)，等号右边所有的项相消，只留下 1:

$$S - AS = 1$$

这就是说

$$S = \frac{1}{1-A}$$

可以用相同的方法计算  $\sum_{i=1}^{\infty} i/2^i$ ，它是一个经常出现的和。我们写成

$$S = \frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \frac{4}{2^4} + \frac{5}{2^5} + \dots$$

用 2 乘之得到

$$2S = 1 + \frac{2}{2} + \frac{3}{2^2} + \frac{4}{2^3} + \frac{5}{2^4} + \frac{6}{2^5} + \dots$$

将这两个方程相减得到

$$S = 1 + \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^5} + \dots$$