HZ BOOKS 华章教育

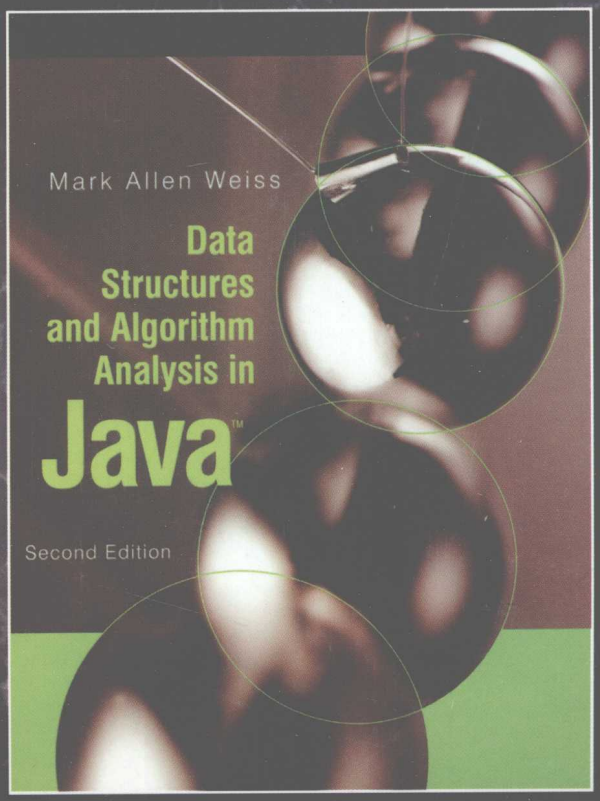
计 算 机 科 学 丛 书

第2版

数据结构与算法分析

Java语言描述

(美) Mark Allen Weiss 佛罗里达国际大学 著 冯舜玺 译



Data Structures and Algorithm Analysis in Java

Second Edition

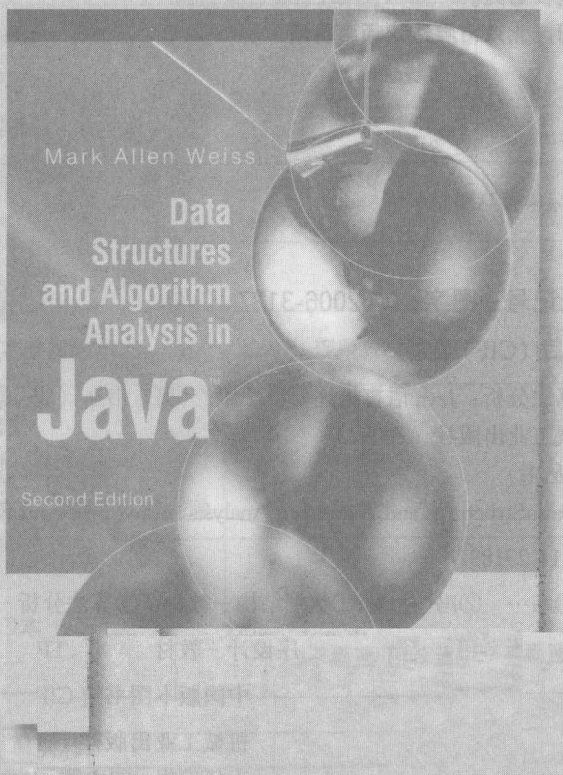
机械工业出版社
China Machine Press

计 算 机 书

第2版

数据结构与算法分析
Java语言描述

(美) Mark Allen Weiss 著 冯舜玺 译
佛罗里达国际大学



Data Structures and Algorithm Analysis in Java
Second Edition

机械工业出版社
China Machine Press

本书是国外数据结构与算法分析课程的标准教材，通俗易懂地介绍了数据结构和算法分析，除讨论一般数据结构及其实现外，还专门讨论了一些高级数据结构及其实现，并在程序代码中充分体现了Java 5.0的新特性。每章末还提供了大量练习，并根据难易程度标记了星级，便于教师、学生使用。本书适合用做高级数据结构课程或研究生算法分析课程的教材。

本书版权登记号：图字：01-2006-3127

图书在版编目（CIP）数据

数据结构与算法分析：Java语言描述　第2版/（美）韦斯（Weiss，M. A.）著；冯舜玺译. —北京：机械工业出版社，2008.3

（计算机科学丛书）

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

ISBN 987-7-111-23183-7

Ⅰ.数…　Ⅱ.①韦…　②冯…　Ⅲ.①数据结构－教材　②算法分析－教材　③Java语言－程序设计－教材　Ⅳ.TP311.12　TP312

中国版本图书馆CIP数据核字（2006）第206006号

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：刘立卿

三河市明辉印装有限公司印刷·新华书店北京发行所发行

2009年1月第2版第1次印刷

184mm×260mm·26印张

标准书号：ISBN 978-7-111-23183-7

定价：55.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

出版者的话

文艺复兴以降，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；也正是这样的传统，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分社较早意识到“出版要为教育服务”。自1998年开始，华章分社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，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum，Bjarne Stroustrup，Brain 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襄助，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，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，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。华章分社欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：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

华章教育

译 者 序

计算机功能的增强、速度的提高和应用的普及，增长了人们对实用算法分析和高效编程实现的需求。在Java语言广泛使用的今天，希望我们这样一本兼顾普及和提高的数据结构与算法分析教材能够对广大读者有所裨益。

本书为《Data Structures and Algorithm Analysis in Java》第2版的中译本。这里，原著者Mark Allen Weiss对第1版进行了全面的修订，将书中的算法、技巧与精心编制的高效Java程序有机地结合起来，通过图示和实例清楚地阐释对每种算法缜密、严格和深入的分析。

应该指出，书中改进最大的方面是用Java 5.0对内容所作的全面更新，尤其是各章的程序。当然，以介绍Java基础为重要内容的第1章发生显著变化则是必然的。再有，第3章对表、栈、队列的讨论已被全面修订。第4章也有些相应的变化，包括对TreeSet类和TreeMap类的讨论。其他各章或多或少都有些相关的更新。众所周知，Java 5.0是Java自发布以来到目前为止改动最大的版本，其强大的新特性和新功能使Java性能产生了巨大的飞跃。因此，本书经过Java 5.0的全面改进，其意义是显而易见的。此外，对前1版中发现的错误，这次第2版均已得到纠正。至于有关第2版更多的信息，读者可从因特网特别是前言中提到的作者Weiss的网站上查到。

在本书翻译过程中，王永柿老师阅读了初稿的大部分章节并提出宝贵的意见和建议，马蒙蒙老师仔细比较并标注了原著第1版和第2版之间的差别，译者衷心感谢他们对翻译工作真诚的帮助。此外，译者特别要感谢广大读者对第1版的深切关爱，并企盼着对本书(第2版)进一步的批评和指正。

译 者

前 言

本书目标

本书新的Java版论述*数据结构*——组织大量数据的方法，以及*算法分析*——算法运行时间的估计。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迫切。可是，由于在输入量很大的时候程序的低效率变得非常明显，因此这又要求对效率问题给予更仔细的关注。通过在实际编程之前对算法的分析，学生可以确定一个特定的解法是否可行。例如，在本书中学生可查阅一些特定的问题并看到巧妙的实现是如何能够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。

一旦确定了解法，接着就要编写程序。随着计算机功能的日益强大，它们必须解决的问题也变得更加庞大和复杂，这就要求我们开发更加复杂的程序。本书的目的是在教授学生良好的程序设计技巧和算法分析能力的同时，使得他们也能够开发出这种极为有效的程序。

本书适用于高级数据结构(CS7)课程或是第一年研究生的算法分析课程。学生应该具有中等程度的程序设计知识，包括面向对象程序设计和递归这样一些内容，此外，还要具有离散数学的一些知识。

处理方法

虽然本书的内容大部分都与语言无关，但是，程序设计还是需要使用某种特定的语言。正如书名指出的，我们为本书选择了Java。

Java是相对较新的语言，它常常用来和C++进行比较。Java具有许多的优点，编程人员常常把Java看成是一种比C++更安全、更具有可移植性并且更容易使用的语言。因此，这使得它成为讨论和实现基础数据结构的一种优秀的核心语言。Java的其他方面，诸如线程和GUI(图形用户界面)，虽然很重要，但是本书并不需要，因此也就不再讨论。

使用Java和C++对数据结构进行的完善描述均在互联网上提供了现成的材料。我们采用类似的编码约定以使得这两种语言之间的对等性更加明显。

本版中最显著的变化

本版(第2版)消除了一些程序中的错误，并对书中的许多部分进行了修订，以使阐述更加清晰。此外：

- 为了体现Java 5.0现代化的特色，我们对书中的程序代码进行了必要的更新。
- 对第3章进行了全面修改，包括对标准`ArrayList`类和`LinkedList`类(以及它们的迭代器)用法的讨论，以及对标准`ArrayList`类和`LinkedList`类的实现的讨论。
- 第4章也得到了修订，包括对`TreeSet`类和`TreeMap`类的讨论，同时用一个宽泛的实例阐述了它们在有效算法设计中的使用。第9章还包括一个例子，利用标准`TreeMap`类来实现最短路径算法。

- 第 7 章包含对一些标准 sort 算法的讨论，包括对在实现重载的标准 sort 算法中所涉及的一些技巧的阐释。

概述

第 1 章包含离散数学和递归的一些复习材料。我相信熟练掌握递归的唯一办法是反复不断地研读一些好的用法。因此，除第 5 章外，递归遍及本书每一章的例子之中。第 1 章还介绍了一些相关知识，作为对基本 Java 的复习和回顾，包括对 Java 5 泛型的讨论。

第 2 章讨论算法分析。本章阐述渐近分析及其主要缺点。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。此外，还介绍了更复杂的分治程序，不过有些分析(求解递推关系)要推迟到第 7 章再进行详细的讨论。

第 3 章包括表、栈和队列。这一章进行了全面的修订。现在的新版包括对 Collections API ArrayList 类和 LinkedList 类的讨论，提供了 Collections API ArrayList 类和 LinkedList 类的一个重要子集的若干实现。

第 4 章讨论树，重点是查找树，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。本章还介绍了 AVL 树和伸展树。查找树实现细节更仔细的处理可在第 12 章找到。树的另外一些内容，如文件压缩和博弈树，推迟到第 10 章讨论。外部介质上的数据结构作为几章中的最后论题来考虑。这一版新增的内容有对 Collections API TreeSet 类和 TreeMap 类的讨论，包括一个重要的例子，描述为求解一个问题而使用三种单独的映射。

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，本章末尾讨论了可扩散列。

第 6 章是关于优先队列的。二叉堆也在这里讲授，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 章论述排序。这一章特别关注编程细节和分析。所有重要的通用排序算法均在该章进行了讨论和比较。此外，还对四种排序算法做了详细的分析，它们是:插入排序、希尔排序、堆排序以及快速排序。本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。这是简短且特殊的一章，如果不讨论 Kruskal 算法则该章可跳过。

第 9 章讲授图论算法。图论算法的吸引力不仅因为它们在实践中经常发生，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 章通过考查一般的问题求解技巧讨论算法设计。该章通过大量的实例而得以强化。这一章及后面各章使用的伪代码使得学生在理解例子时不致被实现的细节所困扰。

第 11 章处理摊还分析，对来自第 4 章和第 6 章的三种数据结构以及本章介绍的斐波那契堆进行了分析。

第 12 章讨论查找树算法、k-d 树和配对堆。不同于其他各章，本章给出了查找树和配对堆完整仔细的实现。材料的安排使得教师可以把一些内容纳入到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和(第 4 章的)AVL 树一起讨论。

第 1 章到第 9 章为大多数一学期的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在前面各章中可以容易地查到。第 9 章里所讨论的 NP-完全性对本书来说太

过简单，你会发现对 NP-完全性再做一些额外的工作以扩充本书内容将是有益的。

练习

每章末尾提供的练习与正文中所述内容的顺序相一致。最后的一些练习是将一章作为一个整体来处理而不是针对特定的某一节来考虑的。难度较大的练习标记一个星号，更难的练习标有两个星号。

参考文献

参考文献列于每章的最后。通常，这些参考文献或者是历史性质的，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。有些文献提供了一些练习的解法。

代码的获得

下面的补充材料在 www.aw.com/cssupport 对所有读者公开：

- 例子程序的源代码

此外，下述材料仅提供给采用本书作为教材的教师。有意者请按照书后所附的“教学支持说明表”中的联系方式联络培生教育出版集团北京代表处。

- 部分练习的解答。
- 来自本书的一些附图。

致谢

在该丛书几部著作的准备过程中，作者得到许多朋友的帮助。有些人在本书的其他版本中曾经提到过，谢谢所有的朋友。

如同往常一样，Addison-Wesley 专家们的努力使得本书写作过程更加轻松。愿借此机会感谢编辑 Michael Hirsch、制作编辑 Marilyn Lloyd。我还想感谢 Paul Anagnostopoulos 和他在 Windfall Software 的同事，感谢他们使最后的散稿成书的出色工作。贤妻 Jill 因其所做的每一件事情应该得到我特别的感谢。

最后，我还想感谢广大的读者，他们发来 E-mail 并指出前面各版中一些错误和矛盾之处。我的网页 http://www.cis.fiu.edu/~weiss 还将包含更新后的源代码(用 Java、C++ 和 C 语言编写)、勘误表以及提交问题报告的一个链接。

M.A.W.
Miami, Florida

目 录

第1章 引 论

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

- 看到程序对于合理的大量输入的运行性能与其在适量输入下运行性能的同等重要性。
- 概括为本书其余部分所需要的基本的数学基础。
- 简要复习递归。
- 概括用于本书的 Java 语言的某些重要特点。

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。“明显的”解决方法是相当多的。

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。当新元素被读到时，如果它小于数组中的第 k 个元素则忽略之，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？使用一千万个元素的随机文件和 $k = 5\,000\,000$ 进行模拟将发现，两个算法在合理的时间量内均不能结束；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，该算法将在一秒钟左右给出问题的解。因此，虽然我们提出的两个算法都能算出结果，但是它们不能被认为是好的算法，因为对于第三种算法能够在合理的时间内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。输入是由一些字母构成的一个二维数组以及一组单词组成。目标是要找出字谜中的单词，这些单词可能是水平、垂直或沿对角线上任何方向放置的。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

现在至少也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。

也可以这样，对于每一个尚未越出谜板边缘的有序四元组(行、列、方向、字符数)我们可以测试是否所指的单词在单词表中。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，假设我们把字谜变成为只给字谜板(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均需要相当长的时间来解决这个问题，从而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，甚至单词表可以很大。

在许多问题当中，一个重要的观念是：写出一个工作程序并不够。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发现需要我们集中精力努力优化的那些代码段。

1.2 数学知识复习

本节列出一些需要记忆或是能够推导出的基本公式，并从推导过程复习基本的证明方法。

1.2.1 指数

$$X^A X^B = X^{A+B}$$

$$\frac{X^A}{X^B} = X^{A-B}$$

$$(X^A)^B = X^{AB}$$

$$X^N + X^N = 2X^N \neq X^{2N}$$

$$2^N + 2^N = 2^{N+1}$$

1.2.2 对数

在计算机科学中，除非有特别的声明，否则所有的对数都是以2为底的。

定义 1.1 $X^A = B$ 当且仅当 $\log_X B = A$。

由该定义可以得到几个方便的等式。

定理 1.1

$$\log_A B = \frac{\log_C B}{\log_C A};\ A, B, C > 0, A \neq 1,\ C \neq 1$$

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，$C^X = B$，$C^Y = A$ 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

$$\log AB = \log A + \log B;\ A, B > 0$$

证明：

令 $X = \log B$，$Y = \log A$，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。

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

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

$$\log X < X \text{ 对所有的 } X > 0 \text{ 成立}$$

$$\log 1 = 0, \log 2 = 1, \log 1024 = 10, \log 1048576 = 20$$

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

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

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

我们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 是其和。此时

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

于是

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

如果我们将这两个方程相减(这种运算只允许对收敛级数进行)，等号右边所有的项相消，只留下 1：

$$S - AS = 1$$

即

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

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

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

用 2 乘之得到

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

将这两个方程相减得到

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

因此，$S = 2$。

分析中另一种常用类型的级数是算术级数。任何这样的级数都可以从基本公式计算其值。

$$\sum_{i=1}^{N} i = \frac{N(N+1)}{2} \approx \frac{N^2}{2}$$

例如，为求出和 $2 + 5 + 8 + \cdots + (3k - 1)$，将其改写为 $3(1 + 2 + 3 + \cdots + k) - (1 + 1 + 1 + \cdots + 1)$，显然，它就是 $3k(k+1)/2 - k$。另一种记忆的方法则是将第一项与最后一项相加(和为 $3k + 1$)，第二项与倒数第二项相加(和也是 $3k + 1$)，等等。由于有 $k/2$ 个这样的数对，因此总和就是 $k(3k+1)/2$，这与前面的答案相同。

现在介绍下面两个公式，不过它们就没有那么常见了。

$$\sum_{i=1}^{N} i^2 = \frac{N(N+1)(2N+1)}{6} \approx \frac{N^3}{3}$$

$$\sum_{i=1}^{N} i^k \approx \frac{N^{k+1}}{|k+1|} \quad k \neq -1$$

当 $k=-1$ 时，后一个公式不成立。此时我们需要下面的公式，这个公式在计算机科学中的使用要远比在数学其他科目中使用得多。数 H_N 叫做调和数，其和叫做调和和。下面近似式中的误差趋向于 $\gamma \approx 0.57\,721\,566$，称为**欧拉常数**(Euler's constant)。

$$H_N = \sum_{i=1}^{N} \frac{1}{i} \approx \log_e N$$

以下两个公式只不过是一般的代数运算：

$$\sum_{i=1}^{N} f(N) = Nf(N)$$

$$\sum_{i=n_0}^{N} f(i) = \sum_{i=1}^{N} f(i) - \sum_{i=1}^{n_0-1} f(i)$$

1.2.4 模运算

如果 N 整除 $A-B$，那么就说 A 与 B 模 N 同余，记为 $A \equiv B \pmod N$。直观地看，这意味着无论是 A 还是 B 被 N 去除，所得余数都是相同的。于是，$81 \equiv 61 \equiv 1 \pmod{10}$。如同等号的情形一样，若 $A \equiv B \pmod N$，则 $A+C \equiv B+C \pmod N$ 以及 $AD \equiv BD \pmod N$。

有许多定理适用模运算，其中有些特别需要用到数论来证明。我们将尽量少使用模运算，这样，前面的一些定理也就足够了。

1.2.5 证明的方法

证明数据结构分析中的结论的两种最常用的方法是归纳法证明和反证法证明(偶尔也被迫用到只有教授们才使用的证明)。证明一个定理不成立的最好的方法是举出一个反例。

归纳法证明

由归纳法进行的证明有两个标准的部分。第一步是证明**基准情形**(base case)，就是确定定理对于某个(某些)小的(通常是退化的)值的正确性；这一步几乎总是很简单的。接着，进行**归纳假设**(inductive hypothesis)。一般说来，它指的是假设定理对直到某个有限数 k 的所有的情况都是成立的。然后使用这个假设证明定理对下一个值(通常是 $k+1$)也是成立的。至此定理得证(在 k 是有限的情形下)。

作为一个例子，我们证明斐波那契数，$F_0=1, F_1=1, F_2=2, F_3=3, F_4=5, \cdots, F_i = F_{i-1} + F_{i-2}$，满足对 $i \geqslant 1$，有 $F_i < (5/3)^i$(有些定义规定 $F_0=0$，这只不过将该级数做了一次平移)。为了证明这个不等式，我们首先验证定理对简单的情形成立。容易验证 $F_1 = 1 < 5/3$ 及 $F_2 = 2 < 25/9$，这就证明了基准情形。假设定理对于 $i=1,2,\cdots, k$ 成立，这就是归纳假设。为了证明定理，我们需要证明 $F_{k+1} < (5/3)^{k+1}$。根据定义得到

$$F_{k+1} = F_k + F_{k-1}$$

将归纳假设用于等号右边，得到

$$\begin{aligned} F_{k+1} &< (5/3)^k + (5/3)^{k-1} \\ &< (3/5)(5/3)^{k+1} + (3/5)^2(5/3)^{k+1} \\ &= (3/5)(5/3)^{k+1} + (9/25)(5/3)^{k+1} \end{aligned}$$

化简后为

$$\begin{aligned} F_{k+1} &< (3/5 + 9/25)(5/3)^{k+1} \\ &= (24/25)(5/3)^{k+1} \\ &< (5/3)^{k+1} \end{aligned}$$

这就证明了这个定理。

作为第二个例子，我们建立下面的定理。

定理 1.3

如果 $N \geqslant 1$，则 $\sum_{i=1}^{N} i^2 = \frac{N(N+1)(2N+1)}{6}$

证明：

用数学归纳法证明。对于基准情形，容易看到，当 $N=1$ 时定理成立。对于归纳假设，设定理对 $1 \leqslant k \leqslant N$ 成立。我们将在该假设下证明定理对于 $N+1$ 也是成立的。我们有

$$\sum_{i=1}^{N+1} i^2 = \sum_{i=1}^{N} i^2 + (N+1)^2$$

应用归纳假设得到

$$\begin{aligned}\sum_{i=1}^{N+1} i^2 &= \frac{N(N+1)(2N+1)}{6} + (N+1)^2 \\ &= (N+1)\left[\frac{N(2N+1)}{6} + (N+1)\right] \\ &= (N+1)\frac{2N^2+7N+6}{6} \\ &= \frac{(N+1)(N+2)(2N+3)}{6}\end{aligned}$$

因此

$$\sum_{i=1}^{N+1} i^2 = \frac{(N+1)[(N+1)+1][2(N+1)+1]}{6}$$

定理得证。 ■

通过反例证明

公式 $F_k \leqslant k^2$ 不成立。证明这个结论的最容易的方法就是计算 $F_{11} = 144 > 11^2$。

反证法证明

反证法证明是通过假设定理不成立，然后证明该假设导致某个已知的性质不成立，从而原假设是错误的。一个经典的例子是证明存在无穷多个素数。为了证明这个结论，我们假设定理不成立。于是，存在某个最大的素数 P_k。令 $P_1, P_2, \cdots, P_k$ 是依序排列的所有素数并考虑

$$N = P_1 P_2 P_3 \cdots P_k + 1$$

显然，N 是比 P_k 大的数，根据假设 N 不是素数。可是，$P_1, P_2, \cdots, P_k$ 都不能整除 N，因为除得的结果总有余数 1。这就产生一个矛盾，因为每一个整数或者是素数，或者是素数的乘积。因此，P_k 是最大素数的原假设是不成立的，这正意味着定理成立。

1.3 递归简论

我们熟悉的大多数数学函数都是由一个简单公式来描述的。例如，我们可以利用公式

$$C = 5(F-32)/9$$

将华氏温度转换成摄氏温度。有了这个公式，写一个 Java 方法就太简单了。除去程序中的说明和大括号外，这一行的公式正好翻译成一行 Java 程序。

有时候数学函数以不太标准的形式来定义。例如，我们可以在非负整数集上定义一个函数 f，它满足 $f(0)=0$ 且 $f(x)=2f(x-1)+x^2$。从这个定义我们看到 $f(1)=1, f(2)=6, f(3)=21$，以及 $f(4)=58$。当一个函数用它自己来定义时就称为是**递归**(recursive)的。Java 允许函数是递归的。[○]

○ 对于数值计算使用递归通常不是个好主意。我们在解释基本概念时已经说过。