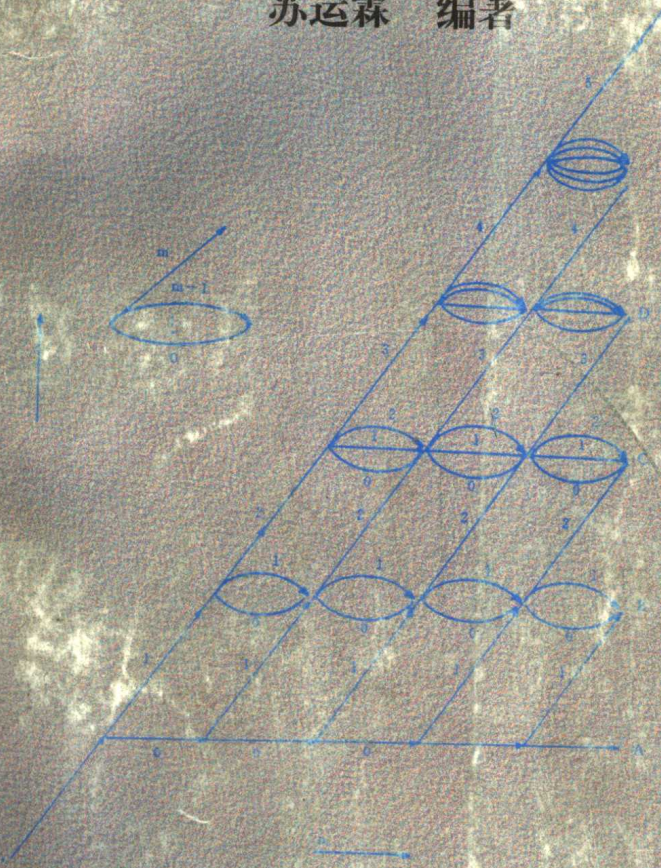
数据结构与算法

苏运霖　编著



中南工业大学出版社

数据结构与算法

苏运霖　编著

中南工业大学出版社
1999·长沙

数据结构与算法

苏运霖　编著

责任编辑：谢盛光

*

中南工业大学出版社出版发行

中南工业大学出版社印刷厂印装

新华书店总店北京发行所经销

*

开本：850×1168　1/32　印张：26　字数：648千字

2000年1月第1版　2000年3月第1次印刷

印数：0001—1000

*

ISBN 7-81061-192-5/O·010

定价：32.00元

内容简介

目前国内尚缺乏反映数据结构与算法的最新发展概况和实用的教材，本书就是为满足该领域的需要而编著的。本书对数据结构作了简略的介绍，用较大的篇幅讲述了必要的数学基础知识，详尽地介绍了当前最新的、实用的算法。本书既注重论述的系统性、逻辑性，又注重文字表达的通俗性、趣味性。该书既是集目前国内外知识前沿之大成，也融会作者多年从事该领域教学与科研之结晶。

该书不仅适用高等院校计算机科学系相应课程的教材和参考书，还可作为科技工作者[redacted]，也可作为自学人士参考之用书。

绪　言

正如算法的设计与分析这门学科的创立者D.E.Knuth所说，算法乃是计算机科学的核心与灵魂，没有算法，就不可能有计算机的程序；而没有程序，任何计算机都不可能有任何作为。因此，在D.E.Knuth的巨著《计算机程序设计技巧》(The Art of Computer Programming)一、二、三卷问世之后，关于算法的研究便如火如荼地开展起来，形成一股热潮。不但新的算法如雨后春笋般地出现，而且新的论著也接二连三地问世。

引为自豪的是，本人是D.E.Knuth的这一不朽巨著的主译者。正是本人，把他的这浩瀚论著，介绍给国人。同时，又连续近10年，从事本领域的教学和科研，因而目睹着算法设计与分析的发展和前进。

D.E.Knuth在《计算机程序设计技巧》第一卷的序言里说了下列这样一段话："……我当然希望这部书将促进对计算机程序设计艺术的进一步的研究，但同时也希望这种促进作用不致过大，以致很快就使这本书本身变得陈腐过时了。"

D.E.Knuth 在这里所表达的交集的感情是不难理解的，因为谁都不希望自己的心血结晶很快地就被历史所淘汰。我们应该说，D.E.Knuth 这一巨著关于算法设计与分析的贡献，将是永垂史册的。然而，这并不意味着，我们不可以推陈出新。在他的著作以及许许多多其他人的著述的基础上，编著出一本更新、更好的论著来，这就是本书写作的动机和目标。

在本人多年从事算法设计与分析的教学与科研的活动中，深感眼下尽管出版了不少算法设计与分析的教材或论著，外文版的数量更多些，但却都不大尽人意。就中文版而言，主要问题是绝大部分都已出版若干年，因而都只能反映直到当时为止的学科成就而无法反映当前的更新成就。其次，尽管这些书都各有千秋，各具特色，但也都使人感到或者描述不够深入浅出，或者分析方面明显不如设计方面，因此两者显得不大平衡。再者，本人认为，对于算法的描述，应尽可能地形式化，使它变得更为优雅，但若做不到，则也不必过于勉强，免得使人感到矫揉造作，又不易理解。最后，也许是最重要的，是 ACM 和 IEEE 于 1991 年推出的联合课程表的建议，对于算法设计与分析这门课的内容和学时，都作了明确的建议，供世界各国的计算机科学系作为参考。其中一点，是建议把这门课开成为数据结构与算法设计与分析的结合，这为今后这门课的教学指明了方向。

因此，按照 ACM 和 IEEE 的建议，编出一本反映当前这门学科的学术成就，也反映其基础的教材，是势在必行的。

本书的内容安排，除了遵循 ACM 和 IEEE 的建议外，还继承了 D.E.Knuth 的做法，首先是为读者提供算法的设计与分析所需要的大部分数学基础。我们这样做，目的是使本书成为自封闭的。同计算机科学的其他课程相比，算法的设计与分析，特别是分析，所要求的数学知识或许可以说是最多的。所以，我们并不限定这里的数学知识就充分了。实际上，数学基础越雄厚，对

于学习算法设计与分析就越有利，因此，可以把这里的数学基础当作是一个中等程度的要求。

在数学基础这部分中，我们讨论了排列和组合、生成函数、递归函数、Fibonacci 数、Catalan 数和 Stirling 数、容斥原理和反演公式，等等。然后，介绍书中经常使用的数学记号。

之后，我们便进入数据结构与算法的正式讨论，按照 ACM 和 IEEE 191 课程表的建议，本书逐次介绍下列内容：

第 2 章，基本数据结构。这一章着重介绍基本数据结构的定义、实现及应用，以及在计算机科学中可找到的相关的运算符。这些包括列表、数组、表格、堆线、树形和图等等。

第 3 章，抽象数据类型。本章介绍抽象数据类型（ADT）的概念，我们把它作为一个方法学引进的动机，并把实现细节同应用分开。

第 4 章，递归算法。介绍在问题求解中递归算法的基础与使用。

第 5 章，复杂性分析。介绍计算复杂性的思想，包括空间复杂性与时间复杂性，并且给出若干实例来说明它在算法分析中的使用。

第 6 章，复杂性类。在这里给出 P 和 NP 的一般概述，包括对于上界、平均和下界的分析。

第 7 章，排序和查找算法。着重对各种排序与查找算法进行比较，从复杂性的角度来评定它们，并且研究时间和空间的折衷。

第 8 章，几何和图算法。介绍几何的表示与操作，包括点、线段和直线的表示，简单性，凸性和相交的判定，以及约丹分类的概念，叙述了合并多边形，求最靠近的点对及求凸外壳等几何算法和求极小跨越树，求最短通路和求极大流等图算法。

第 9 章，问题求解策略。介绍对于构造算法进行问题求解的

不同策略，包括迭代和递归、分而治之、动态规划和贪婪算法。

第 10 章，并行和分布式算法。介绍并行和分布式结构算法的编制。在同串行算法的比较中，说明并行性如何可产生加速。我们既提供了并行算法的例子，也提供了分布式算法的例子，让读者对于这两种类型的算法的特征有感性的认识，并初步掌握编制这两种类型算法的要领。

本书在结构的风格上，参考了国外同类书的做法，即在每章的开始，先介绍该章的目的、动机和提要。而在每章的末尾，有一个小结，然后，再配上适当数量的习题，以让读者自己动手检验自己对课程内容的掌握情况。我们强调读者自己动手动脑多作些题的重要性。这是学好这门课的根本途径。

这本书虽是作者本人编著而成的，但是许多人都为本书作出了贡献，作者首先要感谢本书在撰写过程中所参考的那些书的作者们，是他们的智慧和成果丰富和充实了本书；作者要感谢暨南大学的同事们，因为作者在多年从事本门课程的教学工作中，一直得到他们的支持与不少有益的建议；作者也要感谢学过这门课程的本科生和研究生，他们在学习过程中，也对本课程提出过许多的建议。作者还要对为本书的出版提供极大支持的梁志斌先生以及张团先生表示衷心感谢，也向中南工业大学出版社的童芳远先生和谢贵良先生表示感谢。最后，作者要感谢我的家人对我从事这项艰苦工作所给予的谅解与鼓励，这对本人完成这项工作也是绝对不可少的。

最后，由于本人水平有限，书中错误和缺点在所难免，谨祈读者在阅读过程中，不吝指正。

作者识

1996 年 9 月于暨南园

目　录

第 1 章　数学基础

【内容提要】 在计算机科学的诸多课程中，算法设计与分析这门课程，由于它本身的特殊性，涉及的数学内容是最多、最深的。有人已经为此而写出了专门的书来讨论算法设计和分析所涉及的数学知识。本章对后面数章所涉及的数学内容进行简略的介绍，包括有排列和组合、多项式系数、生成函数、递归函数、Fibonacci 数、Catalan 数和 Stirling 数，容斥原理和反演公式、鸽巢原理和 Ramsey 定理等等。

本章还介绍了在全书中使用的数学符号。

1.1　排列和组合

几乎尽人皆知，n 个对象的排列，就是把 n 个不同的对象放在一行上的一种安排。对于三个对象 1，2 和 3，有六个可能的排列：

123，132，213，231，312，321

在算法分析中，排列的性质十分重要。在后面的章节中，我们将推演它的许多有趣的问题。现在先来考虑 n 个对象有多少种可能的排列。对于排列的最左边位置，显然有 n 种可能的选择，

因为哪一个对象都可排于此处，而一旦完成了这一选择，就有 $n-1$ 种方式来把不同的对象选放在下一个位置上，这样对于头两个位置，就有 $n(n-1)$ 种选择。往下，第三个位置不能和头两个位置相同，因此，可以有 $n-2$ 种选择。以下完全类似。一般说来，如果用 P_{nk} 来表示从 n 个当中选择 k 个对象，并把它们排成一行的方式数，则

$$P_{nk}=m(n-1)\cdots(n-k+1)\quad(k=2,\cdots,n)\qquad(1.1.1)$$

我们可以考虑如何从 $n-1$ 个对象的排列导出 n 个对象的排列。这里介绍两种主要的方法。

方法 1　在 $n-1$ 个对象的每一个排列

$i_1\ i_2\cdots\ i_{n-1}$　其中 $1\leqslant i_j\leqslant n-1, 1\leqslant j\leqslant n-1$

通过在所有可能的位置上插入数 n，以形成 n 个对象的排列，得到

$$ni_1i_2\cdots i_{n-1}, i_1ni_2\cdots i_{n-1},\cdots$$

$$i_1\cdots i_{n-2}ni_{n-1}, i_1i_2\cdots i_{n-2}i_{n-1}n$$

显然，这样就得到了 n 个对象的排列了，对于所有 $n-1$ 的对象的所有排列都这样做的结果，不会产生 n 个对象的同一个排列出现多于一次的情形。

方法 2　这次以下列方法来在 $n-1$ 个对象的每一个排列 $i_1i_2\cdots i_{n-1}$，其中 $1\leqslant i_j\leqslant n-1, 1\leqslant j\leqslant n-1$，形成 n 个其他的排列：

$$i_1i_2\cdots i_{n-1}\frac{1}{2}, i_1i_2\cdots i_{n-1}\frac{3}{2}, i_1i_2\cdots i_{n-1}(n-\frac{1}{2})$$

然后，对于这些排列重新命名，使之成为 $1,2,\cdots,n$ 的一个排列并且具有同样的顺序关系。例如，对于上述的头一个，$\frac{1}{2}$ 就要变为 1，而原来的 1（比如说是 i_l），那它就要成为 2，其余类推，对于上述的第二个排列，$\frac{3}{2}$ 就要变为 2，原来的 1 不动，而原来的 2 要成为

3,其余类推。

显然,通过这一构造,我们也得到 n 个对象的每个排列恰一次。

如果用 P_n 来表示 n 个对象的排列数,即 $P_n = P_{nn}$,则上面两种方法都说明 $P_n = nP_{n-1}$。这也为 $P_n = n(n-1)\cdots 21$ 提供了进一步证明,如同我们已在等式(1.1.1)中建立起来的那样。

这个重要的数量叫做 n 的阶乘,它可写成

$$n! = \prod_{1 \leqslant k \leqslant n} k \tag{1.1.2}$$

我们约定

$$0! = 1 \tag{1.1.3}$$

通过这一约定,对于所有正整数 n,都有基本恒等式

$$n! = (n-1)!n \tag{1.1.4}$$

这里列出头几个阶乘的值

$0! = 1$, $1! = 1$, $2! = 2$, $3! = 6$, $4! = 24$, $5! = 120$,

$6! = 720$, $7! = 5040$, $8! = 40320$, $9! = 362880$

$10! = 3628800$, $11! = 39916800$, $12! = 479001600$

也就是说,10! 大约是 350 万,11! 大约是 4000 万,而 12! 大约是 4.8 亿。在某种意义下,10! 是实际能不能算的近似的分界。如果一个算法需要检验考察多于 10! 的情况,则这就达到不能处理的边缘了。这给了我们什么是在计算机上可行的一个直观的概念。

在 n 的值很大时,$n!$ 的计算变得十分费力。因此,人们长久以来便致力于计算它的近似值。在这方面,最著名的结果是由詹姆斯·斯特林(James Stirling)得出的,这就是

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{1.1.5}$$

符号"≈"表示"近似地等于",而 e 是自然对数的底。例如,我们可以计算

$$40320=8!\approx\sqrt{4\pi}\left(\frac{8}{e}\right)^8=2^{26}\sqrt{\pi}e^{-8}\approx(67108864)\cdot$$
$$(1.77245)(0.00033546)\approx(39902)$$

在这种情况下，误差大约是 1%，可以证明，相对误差近似于 $1/(12n)$。

除了由(1.1.5)式给出的近似值外，也可以较为容易地得到 $n!$ 分解成质因子的精确值。事实上，质数 p 是 $n!$ 的重数为

$$\mu=\left\lfloor\frac{n}{p}\right\rfloor+\left\lfloor\frac{n}{p^2}\right\rfloor+\left\lfloor\frac{n}{p^3}\right\rfloor+\cdots=\sum_{k>0}\left\lfloor\frac{n}{p^k}\right\rfloor \tag{1.1.6}$$

的一个因子。这里 $\lfloor s\rfloor$ 表示 s 的地板函数，即不大于 s 的最大整数，如 $\lfloor 3.5\rfloor=3$，$\lfloor -0.7\rfloor=-1$，如果 $n=1000$，$p=7$，则我们有

$$\mu=\left\lfloor\frac{1000}{7}\right\rfloor+\left\lfloor\frac{1000}{7^2}\right\rfloor+\left\lfloor\frac{1000}{7^3}\right\rfloor=142+20+2=164$$

所以 1000! 能为 7^{164} 所整除，但不能为 7^{165} 所整除。尽管(1.1.6)式被写成一个无穷求和的形式，但对于任何具体的 n 和 p 值，它实际上是有限的，因为后边的所有项总归都是 0。

斯特林的另外一个结果是

$$n!=1+\left(1-\frac{1}{1!}\right)n+\left(1-\frac{1}{1!}+\frac{1}{2!}\right)n(n-1)$$
$$+\left(1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}\right)n(n-1)(n-2)+\cdots \tag{1.1.7}$$

他还发现了一个序列 $a_1,a_2,\cdots$，使得

$$\ln n!=a_1n+a_2n(n-1)+\cdots=\sum_{k\geqslant 0}a_{k+1}\prod_{0\leqslant j\leqslant k}(n-j) \tag{1.1.8}$$

大约在同一时间，伦哈特·欧拉(Leonhard Euler)研究了同一问题，并给出

$$n!=\lim_{m\to\infty}\frac{m^nm!}{(n+1)(n+2)\cdots(n+m)} \tag{1.1.9}$$

这个公式除了对于负整数外(当等式 1.1.9 的分母变成 0 时)的任

意 n 值，都定义了 $n!$。

关于阶乘的另一个重要结果，由艾·玛·勒让德(A.M.Legendre)给出，即

$$n!=\Gamma(n+1)=n\Gamma(n) \tag{1.1.10}$$

函数 $\Gamma(x)$ 叫做伽玛函数，而且由等式(1.1.9)，我们有定义

$$\Gamma(x)=\lim_{m\to x}\frac{m^x m!}{x(x+1)(x+2)\cdots(x+m)} \tag{1.1.11}$$

以上谈的是排列，下边我们开始谈组合。n 个对象每次取 k 个的组合，是从 n 个对象的集合中，取 k 个不同元素的可能的选择。从六个对象 $\{a,b,c,d,e,f\}$ 中每次取三个的组合是

$$abc,abd,abe,abf,acd,ace,acf,ade,adf,aef,$$
$$bcd,bce,bcf,bde,bdf,bef,cde,cdf,cef,def \tag{1.1.12}$$

为了计算 n 个对象取 k 个的组合总数，一个简单的办法是：选择头 k 个对象以形成一个排列的方法有 $n(n-1)\cdots(n-k+1)$ 种；而在这些排列中，每一 k 个元素的组合恰巧出现 $k!$ 次，因为每一组合都出现在它的所有排列中，因此组合的总数记之为 $\binom{n}{k}$，是

$$\binom{n}{k}=\frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots 1} \tag{1.1.13}$$

例如

$$\binom{6}{3}=\frac{6\times5\times4}{3\times2\times1}=20$$

这就是我们在(1.1.12)中找出的组合的数目。

量 $\binom{n}{k}$ 称为二项式系数，这些数的应用格外之多，它们大概是研究算法分析最主要的量，因此，读者应当熟悉它们。

甚至当 n 不是一个整数时，等式(1.1.13)也可用来定义