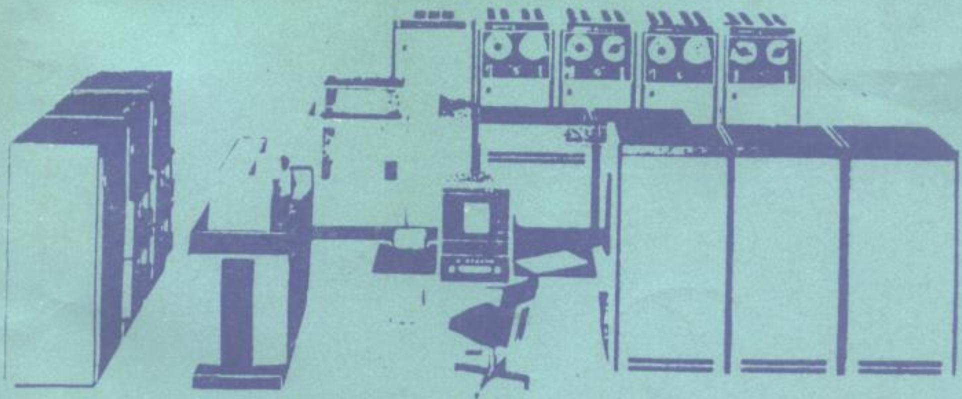
计算机基础知识丛书

计算机科学中的数学基础

施伯乐　楼荣生　蒋家福　编著



科学出版社

内 容 简 介

数学与计算机科学有着紧密的联系．本书比较全面地介绍了计算机科学所需的数学知识，其中包括数制、开关代数、数论、数理逻辑、集合论、递归、概率论、图论、形式语言与自动机等九个方面．

本书内容丰富，叙述由浅入深，注意从例子出发引入和阐述有关理论、概念，特别注重介绍与计算机科学有联系的思想方法，在保持通俗实用的前提下兼顾数学上的系统性与严格性．

本书可供大专院校计算机专业的师生和其他工程技术人员阅读、参考．

计算机基础知识丛书

计算机科学中的数学基础

施伯乐　楼荣生　蒋家福 编著

责任编辑　曾美玉

科学出版社 出版

北京东黄城根北街 16 号

中国科学院印刷厂印刷

新华书店北京发行所发行　各地新华书店经售

1989 年 10 月第 一 版　　开本：787×1092　1/32

1989 年 10 月第一次印刷　　印张：8 5/8

印数：0001—8.850　　字数：193,000

ISBN 7-03-001236-4/TP·76

定价：4.90 元

前　　言

为迎接新的技术革命的到来，一个推广应用计算机的高潮正在全国兴起．广大科技人员、干部、工人和学生都迫切要求学习计算机基础知识．为了适应这一令人鼓舞的形势发展需要，中国计算机学会所属普及委员会决定编写一套《计算机基础知识丛书》，将于近年内陆续出版．本书就是这套丛书之一．

计算机是信息处理的工具．现代计算机的迅猛发展加速了社会信息化的进程． 信息的巨大价值日益被人们所认识．信息的快速生成、广泛传播和有效利用使人类更能高度发挥智力劳动的效益和施展社会组织的才能，从而极大地促使科学技术的进步和推动社会生产力的发展． 普及计算机知识，特别是计算机应用方面的知识，正是整个信息生成、传输、利用过程中不可缺少的重要环节．我们就是基于这样的认识来组织编写《计算机基础知识丛书》的．它应该能为各行各业科技人员提供在自己的专业领域中应用计算机急需的计算机基础知识，同时也为计算机专业科技人员更新和扩展自己的知识领域创造条件．

计算机科学技术是一门综合性很强的学科，发展又十分迅速．我们要求丛书能尽力做到先进性和实用性相结合，科学性和通俗性相结合，硬件和软件相结合．为了保持题材新颖、内容先进，我们采取分批确定选题、随定随写、尽快出版的方针．为了对广大科技人员应用计算机提供实际指导，我们希望能结合一些应用计算机的实例来讲解原理，不但要介绍计

算机有什么用，而且还要介绍它怎样用。我们要求丛书的大多数选题都具有中级科普读物的性质，便于自学，也能提供给相应程度的培训班用作参考教材。作为普及读物，我们自然要更加注意叙述生动形象、文笔通畅易懂，使不同程度的读者都能利用丛书获得收益。

由于我们经验不足，水平有限，疏漏不妥之处在所难免，尚希望读者批评指出，并提出宝贵意见。

《计算机基础知识丛书》编委会

引　言

数学与计算机科学有着密不可分的联系．数学与电子学等学科一起奠定了计算机科学的基础，而计算机科学又推动了数学的发展，为数学开辟了更为广阔的天地．许多成绩卓著的计算机科学家往往同时也是数学家．冯·诺伊曼、图灵等人便是其中的杰出代表．当前，计算机科学已经形成一个拥有庞大队伍，并且新成果层出不穷的重要研究领域．但是，可以不夸大地说，这一领域中每个真正具有划时代意义的成果几乎都同数学有关．“古”今中外，概莫能外．

计算机科学涉及众多的数学分支，穷尽其中每个分支也许都要耗费毕生的精力．我们不可能要求一个计算机工作者精通每个数学分支，却应该期望他们掌握有关数学领域的精华，即主要的思想方法．

本书比较全面地介绍了计算机科学所涉及的一些数学分支的主要内容．对于计算机专业的学生，本书可以弥补现行教材的不足，并帮助他们进一步领会这些数学方法的实质；对于非计算机专业毕业的计算机工作者，本书可以为他们提供关于所需数学知识的概貌．

本书是基本自足的，也就是说阅读时不需要很多前修知识．此外，本书在介绍每个数学分支时不过多追求系统性，而是着重介绍该分支中与计算机科学有关的部分．当然，某些内容具有一定的深度，但经过思考是不难领会的．另外，阅读本书最好和有关的计算机知识互相参照进行．

本书由施伯乐教授组织编写．大纲曾经北京大学杨芙清

教授、清华大学金兰教授审定．全文由复旦大学李为鉴副教授审阅，并经北京工业大学陈祖荫、周鸣方和中国科学院陈传涓复审和修改．在此我们深表感谢．

我们诚恳地希望广大读者和有关专家对本书提出批评指正。

编著者

目　　录

第一章　概　　述

§1.1　计算机科学和技术

计算机科学和技术是一门研究信息和知识的表示、处理、存贮、控制和应用的学科，也是一门涉及较广的理论、方法和技术的学科.

早在19世纪,人们就不断探索着计算机的原理、装置、结构及实现方法. 20世纪40年代，由于电子技术和计算理论的重大进展,第一代计算机于1946年在美国的宾夕法尼亚大学应运而生.

几十年来,计算机事业飞速发展,计算机科学和技术也随之迅速发展：计算机的器件和结构经历了四代，从电子管计算机发展成大规模集成电路计算机；外部设备从只有低速单一功能发展成高速大容量多功能；计算机软件已从机器语言发展成高级语言，并进一步呈现出向非过程性的第四代语言发展的趋势；计算机系统已从仅有机器语言即代码的简单系统发展到硬件、软件、固件三者相结合的系统，并已开始突破冯·诺依曼体系，向着非冯·诺依曼体系发展. 一些国家已开始研究关于知识表示和处理的软硬件系统.

目前,计算机已渗透到国民经济的各个领域,包括人类生活的各个方面. 计算机技术的发展已成为科技进步的主要标志，计算机已成为信息化社会的重要组成部分. 计算机在各行各业广泛地应用,产生了显著的经济效益和社会效益,促进了产业结构、经营管理、服务方式上的巨大变革，本学科与其

它学科的结合，改进和发展了这些学科的研究工具和研究方法，并推动了新型学科的产生. 因此，计算机科学虽然还是一门年轻的学科，但却是发展很快，影响很大的新兴学科. 它已成为整个科学技术领域的带头学科和先导技术.

由于计算机事业的飞速发展，需要研究的课题越来越广泛和深入. 软件工程、数据库、机器翻译、模式识别、计算机图形学、知识库、网络、程序理论、语言理论、计算复杂性等课题的研究都涉及了一定深度的数学知识，如数理逻辑、集合论、概率论、图论、递归函数论、形式语言与自动机等. 本书试图简要地介绍计算机科学和技术研究中必需掌握的数学知识.

§1.2 本书的内容安排

本书分十章，以下几章的内容安排如下:

第二章介绍二进制数的定义、各种运算和数制(二进制与十进制数)的转换. 因为在计算机中是用二进制数来表示、运算的，所以本章是计算机的基础.

第三章介绍开关代数. 开关代数方法大量地应用于计算机的逻辑设计中，从简单的"与门"、"非门"直至加法器的设计.

第四章介绍数论中的整数理论. 它不仅与计算机中数的运算有关，而且也是计算机科学中许多问题的求解基础. 例如，数据结构、算法分析、数据库、软件工程等分支中均要应用数论知识.

第五章介绍数理逻辑初步，主要介绍命题演算和谓词演算. 这些概念在数据库、知识库、人工智能、高级语言中均有广泛的应用.

第六章为集合和关系，主要介绍集合的定义、集合间的关系和运算、集合的归纳定义；并介绍二元关系的性质、等价关

系和划分、偏序和格的定义. 这些内容在计算机科学各分支的理论中都占有一定的地位.

第七章为递归,主要介绍递归的定义、线性递归方程的求解、及组合数. 这些内容在一般的高级语言与形式定义以及算法分析中都将涉及.

第八章为概率论，主要介绍初等概率、随机变量、常用的概率分布、数学期望和方差、伪随机数的产生等内容. 这些内容常出现在计算机科学的一些基础分支（如操作系统、数据结构、数据库、计算机性能评价、软件工程等)中.

第九章为图论,主要介绍图的定义及基本性质、哈密尔顿图、有向图、平面图、对集、图的着色等有关概念和结论. 它们已成为数据结构、算法分析、数据库等方面的理论基础.

第十章为形式语言与自动机，主要介绍形式语言和有限自动机的定义、正规表达式、上下文无关文法等. 这些内容在高级语言、编译原理、自然语言的翻译等方面均有重要的作用.

上述内容在本书中作了浓缩、深入浅出、完整而严格的介绍.

第二章　计算机中的数

数已经是人类的老朋友了.人类很早就与数打上了交道.而对于现代人来说，数已经是一切活动的重要组成部分. 人们为了更有效地与数交往,于是便研制了电子计算机.

但是不要以为计算机中的数与我们普通见到的数是一个样子的，计算机有自己独特的表示数的方法. 本书目的是向读者介绍计算机科学中的数学方法. 作为开始，当然要先阐明计算机中的数是什么样子的.

§2.1　什么是二进制

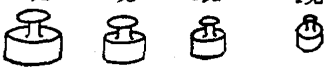
设有一架天平，若两边秤盘中物体的质量差不超过 1/2 克时可用调节平衡杆的方法达到平衡. 问为了要称出 1 到 15 克范围内的物体的质量,至少要配置几个砝码?

显然,只要有整克数的砝码就可以了. 这是因为,对任何一个质量在 1 到 15 克之间的物体,总可以有一个整克数的砝码与它相差不大于 1/2 克. 按照通常的配置方法，只需要配置一个 1 克、两个 2 克、一个 5 克和一个 10 克的砝码. 对于任何一个在 1 到 15 之间的整数,总可以从这 5 个砝码中选出几个来,使它们的总克数恰好就是这个整数.

尽管这种配置方法很符合人们的习惯，并已在许多场合中采用(例如，我国人民币的票面值也是这样配置的)，但它从数学上讲不是最节省的方法. 例如，用四个砝码：一个 1 克的,一个 2 克的,一个 4 克和一个 8 克的，也可以达到同样

目的。图 2.1 就是用这 4 个砝码表示 1 到 15 克共 15 个数的情况，其中 1 表示用到该列顶上的砝码，0 表示不用。如物体质量是 3 克时，用到 1 克和 2 克的两个砝码，而要称质量为 15 克物体时，4 个砝码都要用上。

图 2.1 就是二进制表示的一个实例。每一行是一个二进制数，对应的克数就是与这位二进制数相等的十进制数。



8克	4克	2克	1克	
0	0	0	1	1 克
0	0	1	0	2 克
0	0	1	1	3 克
0	1	0	0	4 克
0	1	0	1	5 克
0	1	1	0	6 克
0	1	1	1	7 克
1	0	0	0	8 克
1	0	0	1	9 克
1	0	1	0	10 克
1	0	1	1	11 克
1	1	0	0	12 克
1	1	0	1	13 克
1	1	1	0	14 克
1	1	1	1	15 克

图 2.1　1—15 的二进制数

仔细观察图 2.1，可以初步了解二进制数的一些简单性质。

每一个二进制数只由 0 和 1 两个数字组成。整数 1 在十进制和二进制中的形式相同，但十进制的整数 2 在二进制中写成 10，而整数 4 与 8 分别在二进制中写成 100 与 1000. 如在图 2.1 中再加上一个 16 克的砝码，则整数 16 在二进制中

写成 10000。一般地，我们可写成：

$$2^n_{(10)} = \underbrace{100\cdots0}_{n\text{个}}{}_{(2)}, \quad n = 0,1,2,\cdots \tag{2.1}$$

这是二进制数与相应十进制数之间的一个最基本的关系，也可看作二进制数的定义，从它可推导出二进制数的另一些性质来。作为本节的结束，我们来计算二进制数的长度。

二进制数给人们带来的一个不好的印象是长。一个不大的十进制数写成二进制数就变成了一长串的 0 与 1. 利用 (2.1) 式，可以计算一个十进制正整数 A 写成二进制后的长度（或位数）。设 n 是满足

$$2^{n-1} \leqslant A < 2^n \tag{2.2}$$

的正整数。在二进制中该不等式写成

$$\underbrace{100\cdots0}_{n-1\text{个}} \leqslant A_{(2)} < 1\underbrace{00\cdots00}_{n\text{个}}$$

其中 $A_{(2)}$ 表示数 A 在二进制中的记法。由上式可见，$A_{(2)}$ 是 n 位的。由 (2.2) 式，n 与 A 的关系是

$$n - 1 \leqslant \log_2 A < n$$

即 $n - 1$ 是 $\log_2 A$ 的整数部分。用符号 $\lfloor x \rfloor$ 表示实数 x 的整数部分，则有

$$n = \lfloor \log_2 A \rfloor + 1 \tag{2.3}$$

如 $A = 15$ 时 $n = 4$. $A = 16$ 时 $n = 5$。

我们知道正整数 A 的十进制位数是 $\lfloor \lg A \rfloor + 1$，所以 (2.3) 式就是在 $\lfloor \lg A \rfloor + 1$ 中将底数 10 改成 2 所得。当 A 很大时，常数 1 及 $\lg A$ 和 $\log_2 A$ 的小数部分都可以忽略不计. 于是 A 的十进制形式和二进制形式的长度之比可近似地表示为

$$\frac{\lg A}{\log_2 A} = \lg 2,$$

即是一个常数。再从一个凑巧的关系

$$2^{10}=1024\doteq 10^3$$

可得：

$$10^n = 10^{3\cdot\frac{n}{3}} \doteq 1024^{\frac{n}{3}} = 2^{\frac{10}{3}n}$$

两边取对数后有：

$$n \doteq \frac{10}{3} n\lg 2 \qquad \lg 2 \doteq \frac{3}{10}$$

这就表明，同一数的二进制和十进制形式的长度比约为10/3。

§2.2 加法与乘法

读者如果在这里是第一次遇到二进制数，那么一定会发觉自己进入了一个陌生而神奇的世界，对于一切都会感到有趣和新鲜。二进制的四则运算就是令人感到有趣的事情之一，它们的运算法则比起十进制要简单得多。

不论是二进制还是十进制，一位数的加法总是其他运算的基础。在二进制中，两个一位数相加只有四种不同的可能，用直式写出来就是

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array},\quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array},\quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array}\quad \text{及}\quad \begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array} \tag{2.4}$$

其中只有第四式产生一个进位。

为了做多位数的加法，除了在个位上可用(2.4)之一相加外，其他各位上都要考虑从低位来的进位，相当于一次要进行三个一位数的相加。现在将三个一位数的加法表按照直式的形式列出如下：

被加数	1	0	1	0	1	0	1	0	
加　数	1	1	0	0	1	1	0	0	
入进位	1	1	1	1	0	0	0	0	(2.5)
和　数	1	0	0	1	0	1	1	0	
出进位	1	1	1	0	1	0	0	0	

式中举出了三个一位数相加的所有 8 种情况．横线上面是相加的三个一位数，分别叫做被加数、加数和从低位来的进位(简称入进位)；横线下面是相加所得的和数以及送往高位的进位(简称出进位)。如加数、被加数都是 1，且低位又有进位时，由第一列查得应有和数 1 并产生一个进位送到高位．事实上，三个 1 之和是 3，即二进制中的 11，个位就是 (2.5) 中的和数，十位就是出进位。

现在设有两个多位数相加，加法从低位向高位逐位进行．首先，在个位上，入进位总是 0，根据加数和被加数的值，在 (2.5) 式后四列之一上查得和数与出进位并写下和数． 这时出进位就是十位数上的入进位．连同十位数上的加数和被加数，又可在 (2.5) 式中查到和数和新的出进位．写下和数，新的出进位参加百位数的加法(作入进位)等等，直到所有位上的数相加完毕．

例 2.1　计算 0101 + 0110．

由下面的直式运算可见，其两数之和为 1011．

被加数	0	1	0	1
加　数	0	1	1	0
入进位	1	0	0	0
和　数	1	0	1	1
出进位	0	1	0	0

尽管我们为了说明原理，在这里用了较多的篇幅，但实际做时是很简单的，绝不会比十进制的加法费力。