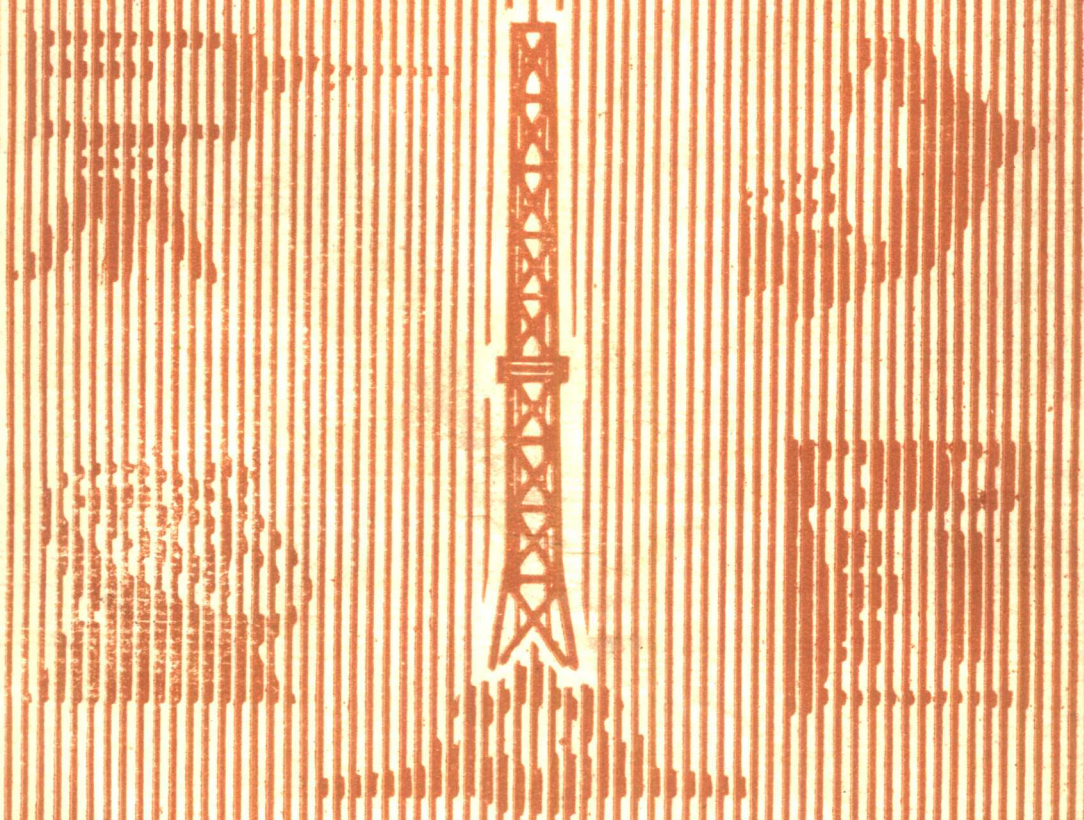


高等学校教材

# 计算机科学导论

贾耀国 朱静华



华中理工大学出版社

高等学校教材

# 计算机科学导论

贾耀国 朱静华

华中理工大学出版社

---

书 名：高等学校教材  
          **计算机科学导论**

---

作 者：贾耀国    朱静华  
责任编辑：陈少华

---

出版发行：华中理工大学出版社（武昌喻家山）  
经 销：新华书店湖北发行所  
印 刷：武汉市新华印刷厂

---

开 本：	787×1092	1/16
印 张：	9.5	
字 数：	202 000	
版 次：	1989年12月第1版	
印 次：	1992年10月第4次印刷	
印 数：	6 501—8 000	

---

ISBN 7-5609-0317-7/TP·26  
定 价：4.50元

---

**(鄂) 新登字第10号**

---

## 出版说明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校、中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978年至1985年，已编审、出版了两轮教材，正在陆续供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻“努力提高教材质量，逐步实现教材多样化，增加不同品种、不同层次、不同学术观点、不同风格、不同改革试验的教材”的精神，我部所属的七个高等学校教材编审委员会和两个中等专业学校教材编审委员会，在总结前两轮教材工作的基础上，结合教育形势的发展和教学改革的需要，制订了1986~1990年的“七五”（第三轮）教材编审出版规划。列入规划的教材、实验教材、教学参考书等近400种选题。这批教材的评选推荐和编写工作由各编委会直接组织进行。

这批教材的书稿，是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会（小组）评选择优产生出来的。广大编审者、各编审委员会和有关出版社为保证教材的出版和提高教材的质量，作出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还会有缺点和不足之处，希望使用教材的单位，广大教师和同学积极提出批评建议，共同为不断提高工科电子类专业教材的质量而努力。

电子工业部教材办公室

## 前 言

本教材系按机械电子工业部的工科电子类专业教材1986~1990年编审出版规划,由《计算机与自动控制》教材编审委员会计算机教材编审小组征稿,推荐出版,责任编委是胡正家教授。

本教材由东南大学担任主编,中国人民解放军通信工程学院谢希仁教授担任主审。

本课程的参考学时数为36学时。本书主要介绍计算机科学的概貌:计算机的发展历史,计算机的技术组成、应用及其对社会的影响等。本书的叙述以算法为中心,并采用类PASCAL语言作为算法的主要描述工具,对算法的设计、表示及实现进行了综合的讨论,特别是对计算机硬件系统和软件系统的各层次作了较详细的介绍。全书共七章:第一章叙述计算机科学的形成及其核心内容;第二章主要讨论算法的描述和分析,并通过具体例子说明算法在计算机上的实现;第三章介绍了布尔代数及其在程序设计语言、组合电路等方面的应用;第四章介绍数据、数据表示、数据结构、数据结构与算法之间的关系;第五章概要叙述不同层次的计算机硬件结构及硬件组织;第六章讨论程序设计语言以及编译程序、操作系统等软件系统;第七章叙述计算机发展及应用,并简要讨论了计算机对社会的影响。

本教材面向计算机类各专业大学一年级学生,也适用于非计算机专业的本科生或研究生。在使用本教材时可根据各专业的不同需要而有所侧重。本教材旨在对计算机科学的概貌作一介绍,它将帮助学生了解计算机专业的性质和任务。但不要期望在学习本教材之后立即能深刻理解书中所提及的所有术语及概念。

本教材第一章、第七章由贾耀国执笔编写,其余各章及附录均由朱静华执笔编写,并由贾耀国统编全稿。参加审阅工作的还有胡鸣钟、叶建勋、杨祥金、沈再福、朱敏、周佩德、滕至阳、黄大海、姜浩等同志。陈行庚、殷新春、汪晖等同志为本书绘图和手稿准备做了大量工作,在此表示诚挚的感谢。由于编者水平有限,书中难免还存在一些缺点和错误,殷切希望广大读者批评指正。

— 编者 —

# 目 录

## 前 言

### 第一章 引论

§ 1.1 计算机科学的由来	( 1 )
§ 1.2 计算机科学的定义	( 2 )
§ 1.3 计算机科学的核心内容	( 3 )
§ 1.4 本书概要	( 5 )

### 第二章 算法

§ 2.1 算法、程序和计算	( 6 )
§ 2.2 算法的描述和分析	( 10 )
2.2.1 欧几里德算法	( 10 )
2.2.2 指令组和流程图表示的欧几里德算法	( 11 )
2.2.3 程序描述	( 12 )
2.2.4 正确性	( 14 )
§ 2.3 在计算机上实现算法	( 16 )
2.3.1 机器语言和欧几里德算法	( 19 )
2.3.2 符号程序设计语言 and 高级语言	( 21 )

### 第三章 布尔代数与应用

§ 3.1 命题逻辑	( 23 )
3.1.1 命题逻辑原理	( 24 )
3.1.2 逻辑的某些应用	( 27 )
§ 3.2 程序设计语言中的布尔表达式	( 27 )
§ 3.3 布尔代数	( 30 )
§ 3.4 开关代数与组合电路	( 33 )
3.4.1 组合电路的生成	( 33 )
3.4.2 操作符集合的完备性	( 36 )

### 第四章 数据和数据结构

§ 4.1 数据	( 38 )
§ 4.2 计算机中数的表示	( 39 )
4.2.1 计算机采用的数制	( 39 )
4.2.2 数的定点表示和浮点表示	( 40 )
4.2.2.1 定点表示法	( 41 )
4.2.2.2 浮点表示法	( 41 )
4.2.3 数的原码表示与补码表示	( 42 )
§ 4.3 数据编码	( 42 )
§ 4.4 数据结构	( 44 )
4.4.1 数组	( 44 )

4.4.2	线性表	( 45 )
4.4.3	非线性数据结构	( 47 )
4.4.3.1	树	( 47 )
4.4.3.2	图	( 47 )
§ 4.5	数据结构与数据类型	( 48 )
4.5.1	数组类型	( 49 )
4.5.2	记录类型	( 50 )
4.5.3	动态结构的实现	( 51 )
§ 4.6	数据结构与算法	( 52 )
§ 4.7	数据与信息	( 54 )

## 第五章 计算机组成和硬件设计

§ 5.1	硬件结构与结构块	( 55 )
§ 5.2	基本电子电路	( 56 )
5.2.1	门	( 57 )
5.2.2	存储元件	( 57 )
5.2.3	时钟	( 59 )
§ 5.3	数字逻辑	( 60 )
5.3.1	组合网络	( 60 )
5.3.2	存储系统	( 62 )
5.3.2.1	寄存器	( 62 )
5.3.2.2	主存储器	( 63 )
5.3.3	时序电路	( 64 )
5.3.3.1	移位寄存器	( 64 )
5.3.3.2	计数器	( 65 )
5.3.3.3	状态机设计	( 67 )
§ 5.4	计算机组织：一个样板机	( 69 )
5.4.1	机器说明	( 69 )
5.4.2	数据流与指令处理	( 73 )
5.4.3	扩充与变化	( 76 )
§ 5.5	联机系统与计算机网	( 77 )

## 第六章 程序设计语言与软件系统

§ 6.1	计算机系统的软件组成部分	( 79 )
6.1.1	装入和执行机器语言程序	( 79 )
6.1.2	符号语言及其处理程序	( 81 )
6.1.3	用于程序编制和执行的软件辅助手段	( 83 )
6.1.4	操作系统的作用	( 83 )
6.1.5	虚拟的机器层次	( 84 )
§ 6.2	程序设计语言	( 84 )
6.2.1	历史的回顾	( 84 )
6.2.1.1	科学计算语言	( 84 )
6.2.1.2	商业数据处理	( 85 )
6.2.1.3	通用语言	( 87 )
6.2.1.4	其他程序设计语言	( 87 )

6.2.2	语法说明	( 87 )
6.2.3	过程与函数	( 92 )
§ 6.3	编译程序	( 94 )
6.3.1	波兰后缀表示法	( 96 )
6.3.1.1	中缀变换成波兰后缀	( 97 )
6.3.1.2	计算波兰后缀	( 99 )
6.3.2	符号表	( 101 )
§ 6.4	操作系统	( 104 )
6.4.1	操作系统的类型	( 105 )
6.4.2	操作系统的功能	( 106 )
6.4.2.1	处理机管理	( 107 )
6.4.2.2	作业管理	( 107 )
6.4.2.3	存储管理	( 108 )
6.4.2.4	文件管理	( 109 )
6.4.2.5	设备管理	( 109 )
6.4.3	操作系统的结构	( 110 )
6.4.4	结语	( 111 )

## 第七章 计算机的发展及应用

§ 7.1	计算机发展简史	( 112 )
7.1.1	古代计算工具	( 112 )
7.1.2	电子计算机的诞生	( 115 )
7.1.3	电子计算机的飞速发展	( 117 )
7.1.4	中国计算机的发展	( 120 )
§ 7.2	人工智能	( 121 )
7.2.1	自然语言理解	( 122 )
7.2.2	专家系统	( 124 )
7.2.3	智能机器人	( 126 )
§ 7.3	计算机与社会	( 128 )
7.3.1	计算机对劳动生产率的影响	( 128 )
7.3.2	计算机对就业的影响	( 129 )
7.3.3	组织机构上的变化	( 129 )
7.3.4	邮政业务	( 130 )
7.3.5	勿需“现金”	( 130 )
7.3.6	报纸、书籍及图书馆	( 130 )
7.3.7	教育方面的变化	( 131 )
7.3.8	工作场所的变化	( 131 )
7.3.9	结语	( 131 )

## 附录 类PASCAL语言 参考文献



# 第一章 引 论

## §1.1 计算机科学的由来

在本世纪中，科学和工程的革命性进展几乎使所有技术领域的内容和方法都发生了巨大的变化，例如物理、生物、医学、电子工程、化学、运输、通讯、纺织、能源和农业等。同时，这些进步和发展也促使一些新领域建立起来，计算机科学就是其中之一。

计算机科学是一门年轻的科学。最早使用“计算机科学”(Computer Science)这个名称的是美国斯坦福大学的G. Forsythe (G.福赛思)。他在1961年发表的一篇论文中写道：“计算机发展如此迅速，以致使计算机科学家也跟不上……，虽然其应用的方面多种多样，但用计算机处理困难问题的方法却表现出很大的统一性，因此，计算机科学这个名称就被加到这个新产生的学科之上”。随着计算机应用的迅速发展，越来越多的人认识到这种统一性；不管计算机用于工程设计、医疗数据处理，还是作曲，其计算方式却是共同的。因此，计算机科学这个名词很快地流行起来，世界上各重要学府随之纷纷设立起以“计算机科学”命名的系科或研究机构，其发展之迅速，在新兴科学中也是较少见的。

从某种意义上讲，计算机科学的起源可以追溯到几百年甚至几千年以前。不过，其中最有意义的是在本世纪40年代所建造的世界第一台全自动的通用电子数字计算机。不言而喻，计算机科学应该研究计算机，因此，在世界上第一台计算机出现之前，这个领域是不存在的，也不可能存在。但是，在很早以前就有了关于计算机硬件概念的阐述。英国数学家C. Babbage (C.拜比吉)在这方面做了突出的工作，他指出了计算机组织的基本原则，并且在19世纪初期设计了一台计算机。

计算机科学的许多理论基础源出于本世纪30年代英国人A. Turing (A.图灵)的研究。他研究了一个通用计算设备的数学模型(即图灵机)，并证明了这个抽象设备的若干引人注目的特性。至今，图灵机仍然被认为是计算机的理论模型。概括地说，计算机科学的近代起源是第一台计算机的建造及计算机数学模型的形成。

历史上，曾经有过许多意义重大的发明，例如汽车、电话等。它们都不能成为独立的科学领域，亦即，并不存在“汽车科学”或“电话科学”。它们只能分别是已有的机械工程、电子工程的一部分。为什么计算机这种机器却能够形成一门独立的、崭新的计算机科学呢？在某种意义上说，如果不熟悉本书所述的某些内容，将很难回答上述问题。不过，我们可以就以下三个方面来概要地加以说明。

第一，从实质上看，计算机科学的必然性来源于计算机本身的特殊性。大家知道，任何机器都是人发明和制造出来的，并且是用来延伸人类本身的器官的。例如，一切交通工具都是人腿功能的延伸；一切工作母机都是人手能力的扩展；望远镜、电子显微镜等又是人眼视觉的延长；电话等则是人耳听觉的伸展等等。计算机呢？计算机并不简单地是某一器官的延伸，它可以说是指挥着这一切器官的器官——大脑的延伸。这个突出的性质，就决定了计算机具备有其它机器所不能比拟的特点，那就是它的深刻性、广泛性、灵活性和渗透性等

等。因此，必须有一门独立的科学来研究，才能回答计算机研制、发展中所提出来的那些十分深刻而又广泛的问题。

第二，从技术上，也就是从计算机所涉及的内容上来看，计算机的工作有点象一台“万能”的问题解答机器。任何问题，只要能够精确地而不是含糊地进行公式化，都可以放到计算机上加以解决。这种把问题化成计算公式求解的方法，称做算法。算法是计算机科学的核心之一。由于算法的普遍性，因而各行各业的人都可以利用计算机来解决各自的问题。这样，就提出了许多技术上的课题。例如：人们应该用什么语言来描述问题，以便让计算机能够理解？对各类不同的问题，需要使用哪些不同的设备？人们应该如何设计新的计算机，以便更好、更快地解决问题？人们应该如何充分利用计算机的所有潜在能力？等等。对以上这些具体问题的研究大致上形成了计算机科学内的硬件设计及软件设计两大部分。与此有关的，还有许多抽象的理论问题，例如，如果说计算机几乎能够解决所有可以精确地加以公式化的问题，那么还有没有计算机所不能解决的问题？如果有的话，它们又是些什么问题？另外，是不是计算机一定能在某一段合理的时间中解决某一特定问题？计算机可以解决很复杂的问题，那么，这些问题可以复杂到什么程度？以上这些问题，就组成了计算机科学中的另一个必须研究的部分，叫做计算理论。另外，如果说计算机是人类大脑的延伸，它是否能完成人类大脑所有的智能？又应该如何让计算机逐步具有人类的智能？等等。这些又组成了计算机科学的另一个研究前沿，即所谓人工智能的研究。计算机科学与数学有相似之处，它可以为各行各业提供一种普遍适用的工具。另一方面，它也具有与工程技术及其它自然科学或社会科学相似之点，那就是它必须从实际出发考虑和解决现实生活中的问题，并且在解决过程中还要融汇许多人类的经验和经验数据。所有这一切技术上的特点，就决定了必须有一门独立的科学来对它加以研究。

第三，从社会作用上来看，计算机对社会政治、经济和文化生活等各个领域的影晌和冲击，其程度之深刻，范围之广泛，是没有一种其它的机器可以与之相比拟的。正因如此，现代社会也被称为“计算机社会”或“信息化社会”。而且，随着计算机的不断发展，它对社会的政治、经济、文化、科学、思想、生活方式等的影晌还在不断扩大、加深。对这样一种影晌如此深远的社会现象，我们当然需要用一门独立的科学来加以研究，以便指导计算机向着造福于人类的正确方向发展。

由于计算机已不同程度地进入到社会的各个方面，因此，每一个接受高等教育的人，无论是工科、理科、管理学科还是文科的学生，都应当对计算机及其实现有一个基本理解。这样，本书既可作为计算机专业学生入门课程的教材，也适用于非计算机专业的本科生或研究生。

## §1.2 计算机科学的定义

在讨论这个问题之前，我们必须承认，目前，用一种很简单的概念概括全部计算机科学的内容，亦即给计算机科学下一个恰当的、精确的定义是非常困难的。一方面是由于计算机科学还很年轻，另一方面也由于计算机科学所涉及的范围非常广泛。

首先，让我们讨论一下计算机科学的总的特点。

大家以往所学过的学科，不论是研究自然现象的物理、化学、天文、地理或生物等学科，也不论是研究社会或人类自身行为等现象的历史、文学、政治、经济或社会学等学科，

它们都有一个共同的特点，那就是对于某一特定范围内客观存在的现象进行观察，并抽象出其内在的联系即科学规律。计算机科学与所有这些科学都不同，它不是去观察已经存在的客观世界并抽象出其规律，而是去研究人类自身所创造的东西——计算机。在这个意义上，它可以被称为是一门人工科学（Artificial Science）。因此，这也就决定了计算机科学中的主要研究方法，不再是象我们学习物理、化学、生物等那样，去观察自然界的事实并发现其规律，而是去分析和设计我们自己的创造物，并改进它的性能。了解这一点是认识和学好计算机科学的一个关键。

计算机科学可以从各种不同的角度来加以定义，我们这里列举下面4种：

#### 1. 以设备为中心

这是一种直截了当的定义。它指出：计算机科学就是研究计算机的科学。

按照这种观点，计算机科学就应该重点研究计算机这一设备本身，以及它的设计、分析、结构和使用等问题。

#### 2. 以信息为中心

这种观点认为计算机科学不应该局限在机器本身，而应该把研究重点放在信息或数据上。它指出：计算机科学是研究信息的科学。

这样一来，计算机科学就应该研究信息或者数据的表示、存储、组织、传送和处理。

#### 3. 以程序为中心

众所周知，程序就是按照一定顺序排列好的一组命令（指令或语句），用来解决某一特定的问题。按照以程序为中心的观点，计算机科学可以定义为：计算机科学是研究程序设计的科学。

根据这种观点，计算机科学就应该研究下述问题：进行程序设计的适当语言是什么？如何证明已设计好的程序是否正确？编写程序有什么技巧？等等。

应该说，所有上述定义在某种意义上都是对的。每一个定义都从各自的侧面出发概括了计算机科学的大部分课题；第1种定义强调的是机器本身；第2种定义则抓住了信息，即机器所处理的对象；第3种则集中在人与机器之间相互通信时所依赖的程序。但是，正因为它们各自强调了不同的方面，所以，就难免不够全面。于是，就有了第4种定义。它是一种比较全面、比较概括、能把上面3种定义都包括的定义。

#### 4. 以算法为中心

这种定义断言：计算机科学是研究算法的科学。

现在，让我们来看一看，这个定义是如何概括上述三种定义的。第1种定义谈机器，而计算机正是具体实现算法的机器；第2种定义谈信息，而信息正是算法处理、操作的对象；第3种定义谈程序，而程序则正是描述算法的工具。因此，我们认为，以算法为中心给计算机科学下定义比较妥当、比较全面。本书将围绕着算法展开讨论。

### § 1.3 计算机科学的核心内容

从理论和实践两方面来看，计算机科学包括技术、应用和社会问题3个部分。这3个部分之间存在着相互联系，其关系示于图1.1中。

技术进步导致应用的发展和社会问题的产生，而应用的需求和社会的压力反过来又大大影响了计算机的技术方面；同样，社会问题与应用开发之间也存在着相互的影响。

虽然计算机科学主要涉及科学和技术问题，但是计算机科学家必须知道上述三个方面的内容及其相互关系。下面，我们将介绍计算机科学在上述三方面中的一些主要内容：

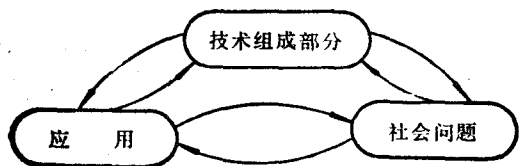


图1.1 计算机科学各课题之间的关系

1. 技术组成部分  
(1) 设计：机器、程序设计语言和系统程序；

(2) 分析：正确性、有效性；

(3) 理论和基础：自动机理论和可计算

性、计算复杂性、形式语言。

2. 一般应用

(1) 科学计算；

(2) 建立模型和模拟；

(3) 工厂自动化、办公室自动化和家庭自动化；

(4) 人工智能。

3. 社会问题

(1) 经济和组织管理体制；

(2) 职业；

(3) 保密；

(4) 按计算机模型制定计划。

应该指出，上述内容不是一成不变的，而且随着各有关领域的发展，新的内容会不断产生，原有内容的含义也会不断变化。

可将技术组成部分概括为算法的设计、实现、分析和理论。计算机科学试图回答下列问题：

(1) 为了有效地执行算法，如何合理地设计机器？

(2) 什么样的语言是一个好的程序设计语言？

(3) 如何设计并实现一个好的算法？

(4) 某一具体的算法是否在所有情况下都正确？

(5) 某一具体算法的效率如何？

(6) 求解某一特定问题的算法是否存在？

(7) 对于某一给定问题，什么样的算法是较好的算法？

(8) 如何定义程序设计语言的语法（形式）和语句（含义）？

在应用领域中也包含计算机科学的研究。与科学计算有关的数学问题的数值求解算法是计算机最早的一种应用。这可以追溯到古代的计算设备。算法的另一个方面是用模拟方法对问题求解（通常是数字的）。通过对被研究的对象建立计算机模型并运行模拟程序，以便对现实世界的未来行为作出预言。为了不断满足各种各样的应用需要，计算机在功能上不断扩大，从科学计算工具、控制工具和经营管理工具扩展到办公工具、学习工具和生活工具，推动着工厂自动化、办公室自动化和家庭自动化的进程。人工智能研究的目标是产生一个计算机系统，该系统表现出：“智力”行为，这个宏伟的、又常常引起争论的目标，在游戏、符号数学、定理证明及自然语言理解方面已形成了相当丰富的课题。

至少有四个与社会有关的问题是计算机科学发展的直接结果：

(1) 由于计算机的推广使用, 社会劳动生产率、银行业务、邮政业务、教育方式以及许多组织体制都将发生变化;

(2) 随着越来越多的任务由计算机自动完成, 对社会产生一些消极的因素, 例如在资本主义社会中可能引起更多的人失业, 也会产生许多积极的结果, 例如可以增加人们的空闲时间;

(3) 由于有了详细记录个人数据的庞大数据库系统, 引起人们对于可能丧失保密性和误用个人数据的极大关注;

(4) 最后, 一方面大型计算机模型给制定计划的机构提供了高质量的信息, 以便作出明智的决定, 另一方面由于模型太复杂, 以致难以理解在模型中的基本假设, 也很难切合实际地解释其结果, 因此, 接受这些模型的计算机输出并使用这种输出也具有相当的危险。

核科学和生物科学已经告诉我们, 科学并不总是有益的。我们必须考虑研究的方向和发展的意义。从根本上来看, 计算机科学也受制于它的社会关系, 科学与社会之间的关系是不可忽视的。

## §1.4 本书概要

本书各章的主要内容示于图1.2中。图中箭头表示阅读的先后顺序。

从第二章起分别介绍算法、布尔代数、数据及数据结构等基础内容。接着是技术核心课题: 机器(硬件)和软件。最后一章将讨论计算机的发展及应用。为了对计算机科学的建立及发展有一个深刻的印象, 我们也讨论了计算机各发展阶段的历史。

在介绍技术基础时, 我们将讨论一些有关的基本算法。为了描述算法, 我们将采用一种半形式语言——类PASCAL语言。用它描写的算法虽不能直接在机器上运行, 但却十分方便和清晰。

本书供大学或专科学校作为学习计算机科学入门课程的教科书, 它旨在让读者了解计算机科学的概貌。书中所介绍的大部分课题都将作为教学计划中的后续课程而进一步详细讨论, 因此, 不要期望在一次阅读后即可深刻理解书中所述的全部内容。

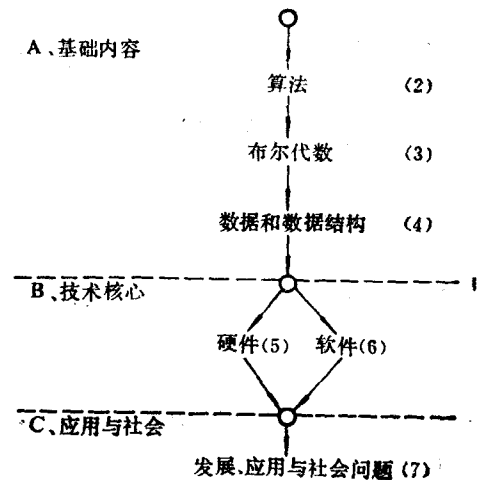


图1.2 本书概要

## 第二章 算 法

所谓“计算机科学的研究以算法为中心”，即认为在计算机科学领域内主要研究这样一些课题：算法的理论、性质；算法的设计和分析方法；描述算法的程序设计语言及其它语言；将算法表示成程序系统及技巧；执行算法的机器（硬件和软件系统）以及计算机科学、技术的社会意义等。在本章中，我们将讨论算法的概念和特性，介绍几种表示算法的方法（包括本书所用的类PASCAL语言在内），并定义一个简单的计算机，以便在该机器上执行算法。我们主要通过设计和研究算法的实例来介绍有关内容。

### §2.1 算法、程序和计算

在实际生活中，我们每天都要解决许多问题。例如，从宿舍去上课走哪一条路？如何完成某课程的作业？如何添置一件新衣？如何安排周末？等等。仔细分析一下就会发现，要解决任何一个问题，首先需要弄清问题的性质、任务和要求，然后再制定切实可行的解决方法。前者是解题过程中的一个重要方面，即对问题进行分析。它是正确、满意地解决问题的基础。问题分析是对问题进行说明的过程，也是找出解题所需全部信息的过程。简单地说，它包括：

- (1) 给出问题的精确说明；
- (2) 列举所有必要的输入；
- (3) 列举所需的全部输出；
- (4) 找出其它有关信息。

在将问题分析完毕后，我们就着手考虑用何种方法来解决它。所谓解决方法就是一组明确的、有一定顺序的步骤，也被称为算法 (Algorithm)。因此，通俗地说，算法是执行特定任务或求解一个具体问题的一组操作步骤（或指令）。

下面举一个简单的例子来说明算法。有三个不同的数，开始时分别放在名为a、b和c的三个盒子中。要求对这三个数进行排序，最后使a中的数小于b中的数，b中的数又小于c中的数。完成此排序任务的算法包括如下步骤：

- (1) 置第1个数于盒子a中；
- (2) 置第2个数于盒子b中；
- (3) 置第3个数于盒子c中；
- (4) 如a中的数大于b中的数，则转向(5)，否则转向(6)；
- (5) 将a与b中的数互换；
- (6) 如b中的数大于c中的数，则转向(7)，否则停止；
- (7) 将b与c中的数互换；
- (8) 如a中的数大于b中的数，则转向(9)，否则停止；
- (9) 将a与b中的数互换；

(10) 停止。

让我们对一组具体的数执行此算法。假定该三个数分别为18、4和11。在执行第1、2、3步之后，第1个数，即18已在盒子a中，相应地，4已在b中且11已在c中。由于18大于4，第4步条件测试结果是真，根据算法应接着执行第5步，即将18与4互换。此后a中的数为4，b中数为18。这样，到了第6步，将b中的数与c中的数进行比较。由于18大于11，而转到第7步，将b与c中的数互换，使b中为11，c中为18。第8步的测试失败（4并不大于11），则计算停止。最后，在盒子a、b和c中的3个数分别为4、11和18，满足了题目的要求，即a中的数小于b中的数，b中的数小于c中的数。不难看出，此算法对任意三个不同的数都能正确工作，即可将它们排序，并按递增的顺序放于a、b和c中。图2.1给出了在所指定的几个执行步骤之后各个盒子中的内容。

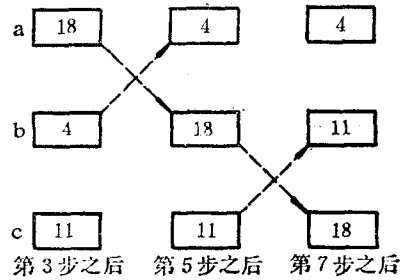


图2.1 排序算法的执行

上述算法是用自然语言描述的，显然这种描述方法并不理想。首先，自然语言有多义性，不易精确描述算法；其次，它也不能清晰地表示算法中的逻辑顺序。另一种表示算法的方法是采用流程图。

图2.2给出了上述排序算法的流程图。若将图中“ $t \leftarrow a$ ”和“ $b \leftarrow a$ ”等概括为“ $x \leftarrow e$ ”，其含义是将值e置入盒子x中。这里要特别注意e的意思：如e为盒子名，则其值为该盒子的内容；如e为代数表达式，例如 $e = (z + 15)/y$ ，则其值即为该代数表达式之值。菱形框中的“ $a > b?$ ”表示将盒a中的内容与b中的内容进行比较，测试前者是否大于后者。

在图2.2中，将两个数互换的操作分解成3条指令。盒子t用来临时存放中间结果。如要互换两个数，如数a和数b（注意：今后将盒a中的数简称为“数a”或“a”），我们首先将一个数，例如是数a，临时存入盒t中（ $t \leftarrow a$ ），其次将另一个数，数b存入盒a中（ $a \leftarrow b$ ），最后再将第一个数从盒t置入盒b（ $b \leftarrow t$ ）。不难理解，仅用“ $a \leftarrow b, b \leftarrow a$ ”不能实现数a和数b的互换。因为在执行了“ $a \leftarrow b$ ”以后，盒a中原来的内容已被破坏。

用计算机的术语说，这种带标号的盒子被称为存储器或存储单元，它们可用于存放数据或其它信息。用流程图表示算法具有下述优点：

（1）算法的逻辑结构明显；  
（2）容易形成算法的模块结构；  
（3）直观易学。

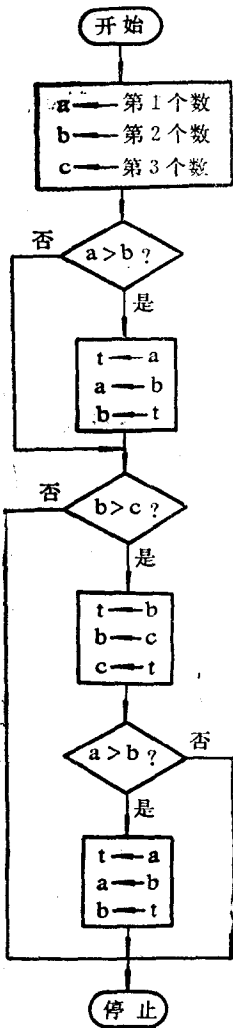


图2.2 排序算法的流程图

但是，我们所设计的算法最终要在计算机上实现，而计算机却比较难于存放并识别这种流

程图。就这一点而言，流程图表示法有不足之处。

为此，我们要用计算机较易接受的程序设计语言 (Programming Languages) ①来描述算法。程序设计语言是形式化的语言，它能精确表示算法。但是，它在许多细节上都有严格要求。例如，绝大多数程序设计语言要求在每个语句后面都要有语句分隔符，程序结束应有结束符等。有的程序设计语言还要求事先说明程序中所用变量(如排序算法中的a, b, c等)的类型，等等。这些都给算法设计带来不便。为了使算法便于阅读、理解、设计和分析，以及易于在计算机上实现，我们将采用一种类似PASCAL的半形式语言来描述算法。

类PASCAL语言'即类似程序设计语言PASCAL的一种语言，它是以PASCAL语言②为基础的一种简化的高级语言。它既不象PASCAL那样形式化，又不象自然语言那样非形式化，因此称它为半形式程序设计语言或伪形式程序设计语言。用它描述算法时，重点突出算法的实质，而使人们暂时避开一些烦琐的语言细节，集中研究算法的基本思想和主要结构。这种用于描述算法的半形式语言称为算法描述语言。虽然不能直接将它描述的算法提交给计算机执行，却能很方便地将其转换成计算机上能实现的高级语言程序。

本书附录提供类PASCAL语言的简介及有关应用实例。下面我们将介绍用类PASCAL语言书写的排序算法，并通过它来说明如何使用这种语言。

**PROCEDURE SORT3** (按递增的次序对3个数排序)

```
BEGIN
  READ (a,b,c) {将三个整数读入a, b和c}
  WRITE(a,b,c) {显示或打印3个数}
  IF a>b THEN
    BEGIN {数a>b, 则a和b互换}
      t:=a, a:=b, b:=t {现在a<b}
    END
    {语句执行后a≤b}
  IF b>c THEN
    BEGIN {数b>c, 则b和c互换}
      t:=b, b:=c, c:=t {现在b<c}
      IF a>b THEN
        BEGIN {数a和b互换}
          t:=a, a:=b, b:=t {现在a<b}
        END
      END
    END
    {a, b, c中的数被排序成a≤b≤c}
  WRITE (a,b,c) {显示或打印排好序的数}
END
```

这种书写形式的算法看起来比较长，那是因为我们插入了许多注解。在括号“{”和“}”之间的部分即为注解。变量a、b和c相当于带标号的盒子。读语句(READ)为算法提供输入，即从外部输入3个数给变量。写语句(WRITE)用作输出，可以在某些外部设备上显示或印出变量值。符号“:=”表示赋值，与图2.2的流程图中所用的箭头“←”有相同的含义，

①程序设计语言又称高级语言，种类很多，都有专门著作介绍，在此不另赘述。

②PASCAL语言也是一种程序设计语言，适合于编制系统软件。



即将符号“: =”右边之值赋给左边的变量。

如果在计算机执行算法时所输入的3个数分别为18、4和11,则在计算过程中两个写语句显示的输出分别是:

18      4      11

及

4      11      18

如果我们自己机械地按照算法步骤执行算法,也可以得到上述结果。这种由人来执行算法的方法称为人工模拟(Hand Simulation),它是开发和检验算法的一种重要手段。由执行者(人或计算机)执行一次算法即称为计算(Computation)。为了便于计算机执行算法而用程序设计语言或半形式语言所表示的算法被称为程序(Program)。所有算法都能被编制成程序,但是,由程序设计语言书写的程序并不一定都是算法。

“算法”这个术语由来已久,它源于9世纪一位著名波斯数学家的名字,后来才与运算过程发生联系。随着数字计算机的发展,“算法”这个词又有了更明确的含义:若规定由计算机来进行计算时,则由计算机执行的、可终止的计算称为算法。算法的性质可归纳为:

(1) 有限性——算法仅包含有限个操作步骤(或指令)。在执行有限步之后,计算一定能终止,并得到解答;

(2) 确定性——算法所含的每条指令含义清楚,仅提供唯一的一种解释,无二义性;

(3) 可行性——每条指令都是可执行的,或者说是有有效的,即在执行者(如计算机)能力范围内,且在有限时间内能正确地执行;

(4) 有一定的输入和输出——每个算法都有一个或多个与特定输入数据有关的输出(或解答)。例如,上述排序算法的输入是3个数值不同的数,而输出是按递增次序排列的这3个数。

经常会遇到一些不满足上述规定的情况,尤其是不能终止的情况。例如,下述指令序列:

(1) 将1置入盒子a中;

(2) 将a的内容加2;

(3) 如a的内容等于10,则停止;

(4) 转向(2)。

其中每一条指令都是明确而有效的。但是,计算不能终止,因为在重复执行指令序列时,a的内容顺次等于1, 3, 5, 7, 9, 11, 13, …。因此,这一组指令不能被称为一个算法。

有的指令本身含义不明确。例如“a的内容除以b的内容”,这里既没有保证 $b \neq 0$ ,又没有进一步说明若 $b = 0$ 将如何处理。有的指令超出了我们力所能及的范围。例如“如果存在一种最佳下棋游戏的快速通用算法,则转向5”。这条指令是无效的,因为没有人知道是否存在这种有效的下棋算法(不仅这种最佳算法是否存在值得怀疑,而且这里也未说明“快速”的含义是什么,所以这一指令是不明确的)。

在其它一些有关计算机的著作中,将专门讨论算法的某些理论问题、实现问题及正确性问题等等。在那些著作中也将给出关于算法的完整形式定义。在本节中,我们仅给出一个相应的非精确定义:算法即是具有4个特性(有限性、确定性、有效性以及有一定输入和输出)的一组指令。