

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

计算机科学概论

Introduction to Computer Science

刘艺 蔡敏 李炳伟 编著

- 与国际接轨的计算机科学教材
- 汇集一线教师的宝贵教改经验
- 内容涵盖计算机科学的方方面面



精品系列

 人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

计算机科学概论

Introduction to Computer Science

刘艺 蔡敏 李炳伟 编著



精品系列

人民邮电出版社

北京

图书在版编目 (CIP) 数据

计算机科学概论/刘艺, 蔡敏, 李炳伟编著. —北京:
人民邮电出版社, 2008.11
21 世纪高等学校计算机规划教材
ISBN 978-7-115-18809-0

I. 计… II. ①刘…②蔡…③李… III. 计算机科学—高等
学校—教材 IV.TP3

中国版本图书馆 CIP 数据核字 (2008) 第 138183 号

内 容 提 要

本书作者在研究 CC 2004 课程体系的基础上, 结合多年科研和教学的经验编排各章内容。全书系统介绍了计算机科学与技术的基本概念、方法和技术, 内容涉及数据的表示与编码、计算机系统结构、计算机硬件组成、操作系统、数据结构与算法、程序设计语言、数据库与文件系统、软件工程、计算机网络和计算机信息系统安全。全书通过大量的图片、图表和演示增强读者对阅读的理解和知识的掌握, 通过范例讲解概念和相关的模型, 通过关键字索引、小结和练习帮助读者巩固自己的学习效果。

本书可作为大学本科“计算机基础”课程教材, 也可供自学人员参考使用。

21 世纪高等学校计算机规划教材

计算机科学概论

-
- ◆ 编 著 刘 艺 蔡 敏 李炳伟
责任编辑 滑 玉
执行编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 15.75
字数: 410 千字 2008 年 11 月第 1 版
印数: 1—3 000 册 2008 年 11 月北京第 1 次印刷

ISBN 978-7-115-18809-0/TP

定价: 29.00 元

读者服务热线: (010) 67170985 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

出版者的话

计算机应用能力已经成为社会各行业最重要的工作要求之一，而计算机教材质量的好坏会直接影响人才素质的培养。目前，计算机教材出版市场百花争艳，品种急剧增多，要从林林总总的教材中挑选一本适合课程设置要求、满足教学实际需要的教材，难度越来越大。

人民邮电出版社作为一家以计算机、通信、电子信息类图书与教材出版为主的科技教育类出版社，在计算机教材领域已经出版了多套计算机系列教材。在各套系列教材中涌现出了一批被广大一线授课教师选用、深受广大师生好评的优秀教材。老师们希望我社能有更多的优秀教材集中地呈现在老师和读者面前，为此我社组织了这套“21世纪高等学校计算机规划教材-精品系列”。

“21世纪高等学校计算机规划教材-精品系列”具有下列特点。

(1) 前期调研充分，适合实际教学需要。本套教材主要面向普通本科院校的学生编写，在内容深度、系统结构、案例选择、编写方法等方面进行了深入细致的调研，目的是在教材编写之前充分了解实际教学的需要。

(2) 编写目标明确，读者对象针对性强。每一本教材在编写之前都明确了该教材的读者对象和适用范围，即明确面向的读者是计算机专业、非计算机理工类专业还是文科类专业的学生，尽量符合目前普通高等教学计算机课程的教学计划、教学大纲以及发展趋势。

(3) 精选作者，保证质量。本套教材的作者，既有来自院校的一线授课老师，也有来自IT企业、科研机构等单位的资深技术人员。通过他们的合作使老师丰富的实际教学经验与技术人员丰富的实践工程经验相融合，为广大师生编写出适合目前教学实际需求、满足学校新时期人才培养模式的高质量教材。

(4) 一纲多本，适应面宽。在本套教材中，我们根据目前教学的实际情况，做到“一纲多本”，即根据院校已学课程和后续课程的不同开设情况，为同一科目提供不同类型的教材。

(5) 突出能力培养，适应人才市场要求。本套教材贴近市场对于计算机人才的能力要求，注重理论技术与实际应用的结合，注重实际操作和实践动手能力的培养，为学生快速适应企业实际需求做好准备。

(6) 配套服务完善，共促提高。对于每一本教材，我们在教材出版的同时，都将提供完备的PPT课件，并根据需要提供书中的源程序代码、习题答案、教学大纲等内容，部分教材还将在作者的配合下，提供疑难解答、教学交流等服务。

在本套教材的策划组织过程中，我们获得了来自清华大学、北京大学、人民大学、浙江大学、吉林大学、武汉大学、哈尔滨工业大学、东南大学、四川大学、上海交通大学、西安交通大学、电子科技大学、西安电子科技大学、北京邮电大学、北京林业大学等院校老师的大力支持和帮助，同时获得了来自信息产业部电信研究院、联想、华为、中兴、同方、爱立信、摩托罗拉等企业和科研单位的领导和技术人员的积极配合。在此，人民邮电出版社向他们表示衷心的感谢。

我们相信，“21世纪高等学校计算机规划教材-精品系列”一定能够为我国高等院校计算机课程教学做出应有的贡献。同时，对于工作欠缺和不妥之处，欢迎老师和读者提出宝贵的意见和建议。

计算机科学是一个充满挑战和发展机遇的年轻学科，而计算机科学概论课程则是这门学科的重要基础。随着计算机在各行各业的广泛应用，很多非计算机专业的课程设置中也把计算机科学概论课程列为公共基础课之一。

既然是基础课的教材，那么本书所设定的读者可以既不具有计算机应用技术，也没有太多的计算机知识。即使是一个对计算机一无所知的人，也能通过学习本书而获取大量的计算机科学的基本知识。如果读者已有一定的计算机应用经验，那就更好，能在本书中发现很多有用的理论知识，提高自己的专业水平。

作为 IT 专业基础课教材，本书力求做到：知识体系完整，覆盖面广，内容翔实，深入浅出。

最重要的一点，本书与国际 IT 教材接轨，符合 ACM 的 CC 2004 课程体系。本书在吸收国内同类课程教学改革经验的基础上，参考了 ACM 推荐的 CS0、CS1、CS2 课程设计，使其从广度上覆盖了计算机科学的主要领域。

本书的结构

本书是为计算机概论/导论课程编写的。该课程将为计算机及其相关专业的本科生勾画出计算机科学体系的框架，为有志于 IT 行业的学生奠定计算机科学知识的基础，铺设进一步深入专业理论学习的桥梁。本书强调概念而不是数学模型和技术细节。全书通过大量的图片、图表和演示增强读者对阅读的理解和知识的掌握；通过范例讲解概念和相关的模型；通过关键字索引、小结和练习帮助读者巩固学习效果。

本书共 11 章，各章的主要内容如下。

- **第 0 章 绪论** 本章从整体上介绍计算机科学的概念和内涵。简要讨论计算机的发展历史，并分析计算机对现代社会的影响。
- **第 1 章 数据的表示与编码** 数据是计算机处理的基本元素，本章介绍数据的表示和运算，以及不同数据类型的编码和存储。
- **第 2 章 计算机系统结构** 本章重点介绍计算机系统结构。首先从系统的角度介绍了计算机系统的层次结构，然后从体系结构的角度分析了系统硬件的结构并重点介绍处理器结构，最后介绍计算机软件系统的结构。
- **第 3 章 计算机硬件组成** 本章介绍计算机硬件系统组成，讲述计算机主要硬件组成部分的原理、结构、实现和发展趋势，并介绍当前主流计算机的常用硬件。
- **第 4 章 操作系统** 本章介绍计算机的操作系统，并阐述操作系统的功能作用、基本组成和运行原理。
- **第 5 章 数据结构与算法** 本章介绍数据结构和算法的概念，以及算法的

描述方法和基本结构，讨论了几种常见的数据结构和常用算法。

- **第6章 程序设计语言** 本章介绍计算机程序设计语言的发展和分类，程序设计语言的基本知识，并介绍了程序编译生成的过程和原理。

- **第7章 数据库与文件系统** 本章讨论数据管理的方法，包括数据库和文件两种方式。着重阐述数据库方法的基本概念、原理，以及文件的基本类型和实现方式。

- **第8章 软件工程** 本章讨论规模化软件开发的方法，包括软件过程、需求分析、软件工程方法、软件质量控制、项目管理等内容。

- **第9章 计算机网络** 本章从计算机网络的形成与发展开始，依次讲述了计算机网络体系结构、局域网设备和 Internet 基本技术及应用。

- **第10章 计算机信息系统安全** 本章从计算机网络信息系统的安全威胁入手，依次讲述了信息安全的核心——数据加密、最常见的安全威胁——计算机病毒、最常见的网络信息安全技术防火墙和入侵检测。

尽管本书包含以上章节内容，但实际的教学进度和授课内容可以灵活确定，因为这要取决于课堂教学的安排及读者对所讨论问题的熟悉程度。教学学时建议安排在 48 ~ 60 学时。

本书的特色

本书的一些特色不仅使得本书与众不同，同时也特别有助于入门者去学习。

概念和知识面

我们始终强调概念要比数学模型更重要，我们认为对概念的理解必然左右对模型的理解。同时，我们还特别注意拓宽读者的知识面，使读者能够从更高的层面上了解计算机科学。

图文并茂

全书有多幅精心设计的图片，这些图片可以帮助读者增进对文字的理解。

教辅

本书为授课老师提供 PPT 课件以及相应的教辅材料（如参考习题解答）。需要者可登录人民邮电出版社教学服务与资源网（<http://www.ptpedu.com.cn>）免费下载。

习题

每章末都有小结和习题。小结是对本章所有关键内容和知识点的简明概括，供复习时参考。习题包括复习题和练习题。

- 复习题：测试本章中所有的要点和概念，帮助学生复习巩固重点内容。
- 练习题：通过课后练习题，检查学生能否运用掌握的概念和知识独立思考，解决问题。


致谢

本书是在作者多年科研和教学基础上编写的，主要参考了作者已发表的文章和著作以及教学中积累的资料。书中还参考了其他中外文教材、资料，由于无法在此一一列举，现谨对这些教材和资料的作者表示衷心的感谢。

参与本书编写的老师还有薛丽敏、蔡青、许大琴、马锐、瞿高峰、彭稳高、吴英、刘藩、吴永逸、洪蕾。

一本书的出版离不开许多人的支持，尤其是这本书。为此，感谢我们的家人和朋友。

由于作者水平有限，书中难免有疏漏和不妥之处，恳请各位专家、同仁和读者不吝赐教，并在此表示特别感谢！



<http://www.liu-yi.net>
2008年8月7日于南京

目 录

第 0 章 绪论1	1.3.5 动态数据的编码.....39
0.1 什么是计算机科学.....1	小结.....40
0.1.1 计算机科学的提出.....1	习题.....41
0.1.2 计算机科学的领域.....2	第 2 章 计算机系统结构44
0.1.3 计算机与计算机模型.....4	2.1 计算机系统的多级层次结构.....44
0.1.4 计算机科学中的经典问题.....4	2.2 体系结构的基本概念.....47
0.2 计算机的历史.....8	2.2.1 计算机体系结构.....47
0.2.1 计算机前传.....8	2.2.2 计算机组成.....48
0.2.2 现代计算机时代.....9	2.2.3 计算机实现.....49
0.2.3 计算机的未来.....12	2.3 冯·诺依曼结构和哈佛结构.....51
0.3 我国计算机的发展.....13	2.3.1 冯·诺依曼理论.....51
0.4 计算机与社会问题.....14	2.3.2 冯·诺依曼体系结构.....52
0.4.1 依赖与控制.....14	2.3.3 冯·诺依曼结构的演变.....53
0.4.2 对生活方式影响.....14	2.3.4 哈佛结构.....54
0.4.3 计算机伦理.....15	2.4 处理器体系结构.....55
0.4.4 计算机犯罪.....16	2.4.1 指令系统.....55
小结.....17	2.4.2 CISC 结构.....56
习题.....17	2.4.3 RISC 结构.....58
第 1 章 数据的表示与编码20	2.4.4 流水线技术.....59
1.1 数和数制.....20	2.4.5 并行处理结构.....60
1.1.1 数字系统.....20	2.5 计算机软件系统.....62
1.1.2 计数与进制.....22	小结.....64
1.1.3 二进制和位.....23	习题.....65
1.1.4 八进制和十六进制.....24	第 3 章 计算机硬件组成68
1.1.5 不同进制数的相互转换.....24	3.1 计算机硬件系统组成.....68
1.2 数值的表示与运算.....28	3.2 处理器.....69
1.2.1 整数的表示.....28	3.2.1 运算器.....69
1.2.2 实数的表示.....31	3.2.2 控制器.....71
1.2.3 位的算术运算.....32	3.2.3 摩尔定律与处理器的发展.....71
1.3 非数值信息的编码.....34	3.3 存储器.....73
1.3.1 字符的编码——ASCII.....34	3.3.1 计算机的多级存储系统.....73
1.3.2 汉字的编码.....36	3.3.2 主存储器和高速缓存.....74
1.3.3 Unicode 码.....37	3.3.3 辅助存储器.....76
1.3.4 静态图像的编码.....37	3.4 输入/输出设备.....78

3.4.1 I/O 设备的分类·····	79	5.4 基本算法·····	123
3.4.2 常见输入设备·····	79	5.4.1 排序·····	124
3.4.3 常见输出设备·····	80	5.4.2 查找·····	126
3.5 总线与接口·····	82	5.5 递归·····	127
3.5.1 总线的分类·····	82	小结·····	129
3.5.2 系统总线·····	82	习题·····	129
3.5.3 I/O 总线·····	83	第 6 章 程序设计语言 ·····	133
小结·····	84	6.1 计算机语言的发展·····	133
习题·····	85	6.1.1 自然语言与形式语言·····	134
第 4 章 操作系统 ·····	88	6.1.2 机器语言与汇编语言·····	135
4.1 操作系统概述·····	88	6.1.3 高级语言·····	136
4.1.1 操作系统的基本概念·····	88	6.2 程序设计语言的几种范型·····	137
4.1.2 操作系统的功能作用·····	88	6.3 程序设计语言的语法元素和功能划分·····	139
4.1.3 操作系统的组成·····	90	6.4 程序的生成和运行·····	143
4.1.4 操作系统的发展与分类·····	91	6.4.1 编译和解释·····	143
4.2 操作系统的运行·····	93	6.4.2 程序设计环境·····	146
4.2.1 处理器管理与进程调度·····	93	小结·····	148
4.2.2 存储管理·····	99	习题·····	149
4.2.3 文件系统·····	100	第 7 章 数据库与文件系统 ·····	152
4.2.4 设备管理·····	102	7.1 数据库的基本概念·····	152
4.3 典型操作系统·····	103	7.1.1 什么是数据库·····	153
4.3.1 Windows 系列·····	104	7.1.2 数据模型·····	156
4.3.2 UNIX·····	104	7.1.3 数据字典·····	157
4.3.3 自由软件和 Linux 操作系统·····	105	7.2 关系数据库及其运算·····	158
小结·····	107	7.2.1 关系模型·····	158
习题·····	108	7.2.2 结构化查询语言·····	161
第 5 章 数据结构与算法 ·····	111	7.3 文件系统·····	163
5.1 概述·····	111	7.3.1 文件的基本概念·····	164
5.1.1 数据结构·····	112	7.3.2 文件结构及存取方式·····	164
5.1.2 算法·····	113	7.3.3 顺序文件·····	167
5.2 线性结构·····	117	7.3.4 索引文件·····	167
5.2.1 线性表·····	117	7.3.5 散列文件·····	168
5.2.2 栈和队列·····	118	7.3.6 二进制文件和文本文件·····	169
5.2.3 串·····	119	小结·····	169
5.2.4 数组·····	119	习题·····	170
5.3 非线性结构·····	120	第 8 章 软件工程 ·····	173
5.3.1 树·····	120	8.1 软件与软件工程·····	173
5.3.2 图·····	122		

8.1.1 什么是软件	173	9.2.2 OSI/RM	203
8.1.2 软件工程	174	9.2.3 TCP/IP 模型	204
8.2 软件开发过程	175	9.3 组建网络	206
8.2.1 软件生存周期	176	9.3.1 计算机网络拓扑	206
8.2.2 软件过程模型	176	9.3.2 网络的物理组成	207
8.2.3 需求分析	179	9.3.3 连接网络	208
8.3 软件工程的方法	181	9.4 Internet 及应用	209
8.3.1 结构化方法	181	9.4.1 Internet 简介	209
8.3.2 面向对象的方法	185	9.4.2 传输协议与地址	210
8.4 软件质量管理	188	9.4.3 Internet 应用	213
8.4.1 软件质量的含义	188	9.4.4 Internet 发展新趋势	216
8.4.2 软件质量的评价	188	小结	217
8.4.3 如何保证软件质量	189	习题	217
8.4.4 软件评审	189	第 10 章 计算机信息系统安全	220
8.4.5 软件测试	189	10.1 计算机信息系统安全概述	220
8.5 软件项目管理	191	10.1.1 安全威胁	221
8.5.1 组织计划	191	10.1.2 安全目标	222
8.5.2 资源管理	192	10.1.3 安全服务与安全机制	222
8.5.3 文档管理	192	10.1.4 安全策略	224
8.5.4 项目管理过程	192	10.2 数据加密	225
小结	194	10.2.1 密码学基础	225
习题	195	10.2.2 加密技术	226
第 9 章 计算机网络	198	10.3 计算机病毒	228
9.1 什么是计算机网络	198	10.3.1 计算机病毒概论	229
9.1.1 计算机网络的历史	198	10.3.2 计算机病毒的作用机理	230
9.1.2 计算机网络的定义	199	10.3.3 计算机病毒防治	232
9.1.3 计算机网络的功能	200	10.4 防火墙与入侵检测	233
9.1.4 计算机网络的分类	200	10.4.1 防火墙	233
9.1.5 以太网	201	10.4.2 入侵监测技术	235
9.2 计算机网络体系结构	202	小结	238
9.2.1 概述	202	习题	239

第 0 章

绪论

本章从整体上介绍计算机科学的概念和内涵。简要介绍计算机的发展历史，并分析计算机对现代社会的影响。

本章学习要求：

- 理解计算机科学的发展及其含义；
- 了解计算机科学涵盖的内容；
- 了解计算机的发展及现代计算机的分类；
- 理解计算机与社会问题的联系。

计算机科学是在现代电子计算机发明以后，随着计算机技术的发展和应用的广泛逐渐形成的一门新兴学科。与传统领域相比，它的发展非常迅速，其影响几乎涉及我们社会的所有层面。

0.1 什么是计算机科学

计算机科学是研究计算机及其周围各种现象和规律的科学，即研究计算机系统结构、程序系统（即软件）、人工智能以及计算本身的性质和问题的学科。计算机是一种进行算术和逻辑运算的机器，而且对于由若干台计算机联成的系统而言还有通信问题，并且处理的对象都是信息，因而也可以说，计算机科学是研究信息处理的科学。

0.1.1 计算机科学的提出

自计算机发明以来，曾经围绕着计算机科学能否独立地成为一门学科产生过许多争论。最早的计算机科学学位课程是由美国普渡大学于 1962 年开设的。随后斯坦福大学也开设了同样的学位课程。但针对计算机科学这一名称在当时引起了激烈的争论。毕竟当时的计算机主要用于数值计算，因此大多数科学家认为使用计算机仅仅是编程问题，不需要做深刻的科学思考，没有必要设立学位。当时很多人认为计算机从本质上说是一种职业而非学科。

20 世纪 80 年代计算技术得到了迅猛的发展并开始深入到大多数学科领域，但以往激烈的争论仍在继续。针对激烈的争论，1985 年春 ACM 和 IEEE-CS 联手组成攻关组开始了对计算作为一门学科的存在性证明，经过近 4 年的工作，ACM 攻关组提交了计算作为一门学科 Computing as a Discipline 的报告，完成了这一任务。该报告的主要内容刊登在 1989 年 1 月的 ACM 通信（Communications of the ACM）杂志上。



知识扩展

ACM (Association for Computing Machinery) 国际计算机组织

ACM 是一个国际科学教育计算机组织,它致力于发展在高级艺术、最新科学、工程技术和应用领域中的信息技术。它强调在专业领域或在社会感兴趣的领域中培养、发展开放式的信息交换,推动高级的专业技术和通用标准的发展。

ACM 创立于 1947 年,是世界上第一个,也一直是最大的科学教育计算机组织。其创立者和成员都是数学家和电子工程师。成立这个组织的初衷是为了使计算机领域和新兴工业的科学家和技术人员能有一个共同交换信息、经验知识和创新思想的场合。

计算机科学领域的最高荣誉是 ACM 设立的图灵奖,被誉为是计算机科学的诺贝尔奖。它的获得者都是本领域最为出色的科学家和先驱。华人中首获图灵奖的是姚期智先生,他于 2000 年以其对计算理论做出的诸多“根本性的、意义重大的”贡献而获得这一崇高荣誉。

ACM 对计算机科学是这样定义的:计算机科学(计算学科)是对描述和变换信息的算法过程的系统研究,包括它的理论、分析、设计、有效性、实现和应用。他们认为,全部计算科学的基本问题是“什么能够(有效地)自动进行”。今天,计算机科学一词是一个非常广泛的概念。尽管如此,在本书里,我们将其定义为“和计算机相关的问题”。

计算机科学是一门实用性很强、发展极其迅速的面向广大社会的学科,它建立在数学、电子学(特别是微电子学)、磁学、光学、精密机械等多门学科的基础之上。但是,它并不是简单地应用某些学科的知识,而是经过高度综合形成一整套有关信息表示、变换、存储、处理、控制和利用的理论、方法和技术。计算机科学包含各种各样与计算和信息处理相关主题的系统学科,从抽象的算法分析、形式化语法等,到更具体的主题如编程语言、程序设计、软件和硬件等。作为一门学科,它与数学、计算机程序设计、软件工程和计算机工程等学科之间存在不同程度的交叉和覆盖。

0.1.2 计算机科学的领域

计算机科学的几个分支领域包括:数值和符号计算、算法和数据结构、体系结构、操作系统、程序设计语言、软件方法学和工程、数据库和信息检索、计算理论、人工智能和机器人学等。

1. 数值和符号计算

数值和符号计算研究的是有效和精确地求解由数学模型所导出方程的一般方法。基本问题包括:怎样才能按照给定精度很快地解出给定类型的方程;怎样对方程进行符号运算,例如积分、微分和化简为最小项等;怎样把这些问题的回答加入到有效的、可靠的、高质量的数学软件包中去。

2. 体系结构

体系结构主要研究的是将硬件和软件组织成有效和可靠系统的方法。基本问题包括:什么是在一个机器中实现数据处理、存储和通信的最好方法?如何设计和控制大型计算系统并且使它们能够在有错误和故障的情况下完成预期的工作?什么类型的体系结构能使许多处理单元有效地协同工作,实现并行计算?怎样测试度量计算机的性能?

3. 操作系统

操作系统研究的是允许多种资源在程序执行中有效配合的控制机制。基本问题包括:在计算

机系统运行的各级上可见的对象和允许的操作是什么；每一类资源允许有效使用的最小操作集是什么；怎样组织接口，使得用户只处理资源的抽象形式，而可以不管硬件的实际细节；对作业调度、存储器管理、通信、软件资源存取、并发任务间的通信、可靠性和安全的有效控制策略是什么；系统应该在什么功能上扩展；怎样组织分布式计算，使得许多由通信网络连接起来的独立的计算机能够参与同一计算。

4. 数据结构和算法

数据结构和算法主要研究的是一些特定类型的问题及相对应的数据结构和解决方法。基本问题包括：对给定类型的问题，最好的算法是什么；它们要求多少存储空间和时间；空间与时间的折衷方案是什么；存取数据最好的方法是什么；最好算法的最坏情况是什么；算法的运行（按平均来说）好到何种程度；算法一般化到何种程度，即什么类型的问题可以用类似的方法处理。

5. 程序设计

程序设计研究的是执行算法的虚拟机的符号表达、算法和数据的符号表达以及从高级语言到机器码的有效的翻译。基本问题包括：由一种语言给出的虚拟机的可能的组织（数据类型、运算、控制结构、引入新类型和运算的机制）是什么；这些抽象怎样在计算机上实现；用什么样的符号表达（语法）可以有效地指明计算机应该做什么。

6. 软件工程

软件工程研究的是满足技术要求，安全、可靠、可信的程序和大型软件系统的设计。基本问题包括：在程序和程序设计系统的开发背后的原理是什么；怎样去证明程序或系统满足它的技术要求；怎样给定技术要求，使之不遗漏重要的情况，而且可以分析它的安全性；怎样使软件系统通过不同阶段不断改进；怎样将软件设计得易理解和易修改。

7. 数据库和信息检索

数据库和信息检索研究的是对大量持续的分享的数据集合的组织，使之能够进行有效的查询和刷新。基本问题包括：用什么样的模型化概念去表示数据元和它们之间的关系；怎样把存储、定位、匹配、检索等基本操作组合成有效的事务处理；这些事务处理怎么与用户有效地交互作用；怎样把高级查询翻译成高性能的程序；什么样的系统结构能有效地检索和刷新；怎样保护数据，以抵制非法存取，泄露或破坏；怎样保护大型数据库不会由于同时刷新而导致不相容；当数据分散在许多台计算机时，怎样使安全保护和访问性能二者得以兼顾；怎样索引和分类正文，以达到有效的检索。

8. 人工智能和计算理论

人工智能和计算理论研究的是动物和人类（智能）行为模型。基本的问题包括：摹本的行为模型是什么，我们怎样建造机器来模拟；由规则赋值、推理、演绎和模式计算所描写的智能可以达到什么程度；由这些模型模拟行为的机器最终能达到什么性能；感知的数据应如何编码，使得类似的模式有类似的码字；驱动码怎样和感知码相联系；学习系统的体系结构如何，以及这些系统如何表示它们对外部世界的知识；怎样才能用有穷的离散过程去精确地逼近连续或无穷的过程；怎么处理逼近导致的误差等。



读者在阅读本章时可能会发现某些概念或术语比较生僻，这些概念和术语我们将在以后的章节逐个讲述。

注意

0.1.3 计算机与计算机模型

阿兰·图灵在 1937 年首次提出了一个通用的计算设备的设想。他设想所有的计算都可能在一台特殊的机器上执行，这就是现在所说的图灵机。尽管图灵对这样一种机器进行了数学上的描述，他还是更有兴趣关注计算的哲学定义，而不是建造一台真实的机器。他将该模型建立在人们进行计算过程的行为上，并将这些行为抽象到用于计算的机器的模型中，这才真正地改变了世界。

通用图灵机是对现代计算机的首次描述只要提供了合适的程序，该机器就能做任何运算。可以证明，一台很强大的计算机和通用图灵机一样能进行同样的运算。我们所需要的仅仅是提供这两者数据以及说明如何做运算的程序。实际上，通用图灵机能做任何可计算的运算。

基于通用图灵机建造的计算机都是在存储器中储存数据。在 1944 ~ 1945 年期间，冯·诺依曼指出，鉴于程序和数据在逻辑上是相同的，因此程序也能储存在计算机的存储器中。基于冯·诺依曼模型建造的计算机有 4 个子系统：存储器、算术/逻辑单元、控制单元和输入/输出单元。冯·诺依曼模型中要求程序必须存储在存储器中。这和早期只有数据才存储在存储器中的计算机结构完全不同。现代计算机的存储单元主要用来存储程序及其响应数据。这意味着数据和程序应该具有相同的格式，因为它们都储存在存储器中。实际上它们都是以位模式（0 和 1 序列）存储在内存中的。

计算机科学的大部分研究是基于“图灵机模型”和“冯·诺依曼模型”的，它们是绝大多数实际机器的计算模型。作为此模型的开山鼻祖，邱奇—图灵论题（Church-Turing Thesis）表明，尽管在计算的时间，空间效率上可能有所差异，现有的各种计算设备在计算的能力上是等同的。尽管这个理论通常被认为是计算机科学的基础，可是科学家也研究其他种类的机器，如在实际层面上的并行计算机和在理论层面上概率计算机和量子计算机等。在这个意义上来讲，计算机只是一种计算的工具。著名的计算机科学家 Dijkstra 有一句名言“计算机科学之关注于计算机，并不甚于天文学之关注于望远镜”。

0.1.4 计算机科学中的经典问题

在人类社会的发展过程中，人们提出过许多具有深远意义的科学问题，其中一些问题对计算学科一些分支领域的形成和发展起了重要的作用。另外，在计算学科的发展过程中，为了便于对计算学科中有关问题和概念的本质的理解，人们还给出了不少反映该学科某一方面本质特征的经典实例。计算学科典型问题的提出及研究不仅有助于我们深刻地理解计算学科，而且还对该学科的发展有着十分重要的推动作用。下面分别对图论中有代表性的哥尼斯堡七桥问题算法与算法复杂性领域中有代表性的梵天 Hanoi（又译为汉诺塔问题）算法复杂性中的难解性问题，哲学家共餐等问题及其相关内容进行分析。

1. 哥尼斯堡七桥问题

17 世纪的东普鲁士有一座哥尼斯堡（Konigsberg）城，现为俄国的加里宁格勒（Kaliningrad）城，城中有一座奈佛夫（Kneiphof）岛。普雷格尔 Pregol 河的两条支流环绕其旁，并将整个城市分成北区，东区，南区，和岛区 4 个区域。全城共有 7 座桥，将 4 个城区相连起来，如图 0-1 所示。人们常通过这 7 座桥到各城区游玩，于是产生了一个有趣的数学难题：寻找走遍这 7 座桥，且只许走过每座桥一次最后又回到原出发点的路径。该问题就是著名的哥尼斯堡七桥问题。

1736年,大数学家列昂纳德·欧拉(L.Euler)发表了关于哥尼斯堡七桥问题的论文——《与位置几何有关的一个问题的解》(Solutio Problematis ad Geomertriam Situs Pertinentis)。他在文中指出,从一点出发不重复地走遍七桥最后又回到原出发点是不可可能的。为了解决哥德斯堡7座桥问题,欧拉用4个字母ABCD代表4个城区并用7条线表示7座桥。如图0-2所示。

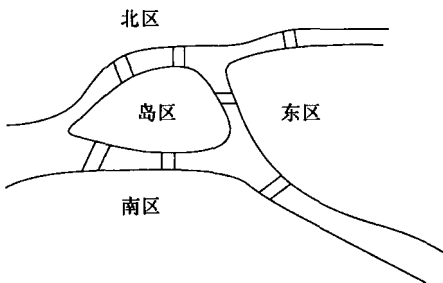


图 0-1 哥尼斯堡地图

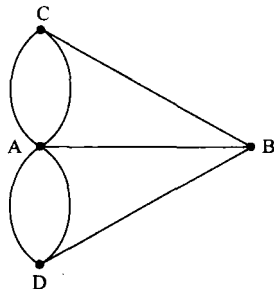


图 0-2 欧拉图

可以看到,在图0-3中,只有4个点和7条线。这样做是基于该问题本质考虑的。它抽象出问题最本质的东西,忽视问题非本质的东西如桥的长度等,从而将哥尼斯堡七桥问题抽象为一个数学问题,即经过图中每边一次且仅一次的回路问题了。欧拉在论文中论证了这样的回路是不存在的,后来人们把有这样回路的图称为欧拉回路或欧拉图。

欧拉在论文中将问题进行了一般化处理,即对给定的任意一个河道图与任意多座桥,判定可能不可能每座桥恰好走过一次,并用数学方法给出了3条判定的规则。

- (1) 如果通奇数座桥的地方不止两个,满足要求的路线是找不到的。
- (2) 如果只有两个地方通奇数座桥,可以从这两个地方之一出发,找到所要求的路线。
- (3) 如果没有一个地方是通奇数座桥的,则无论从哪里出发所要求的路线都能实现。

欧拉的论文为图论的形成奠定了基础,今天图论已广泛地应用于计算学科运筹学、信息论、控制论等学科之中,并已成为我们对现实问题进行抽象的一个强有力的数学工具。随着计算学科的发展,图论在计算学科中的作用越来越大,同时图论本身也得到了充分的发展。

在图论中还有一个很著名的汉密尔顿回路问题。该问题是爱尔兰著名学者威廉·汉密尔顿爵士(W.R.Hamilton)于1859年提出的一个数学问题。其大意是问在某个图G中能不能找到这样的路径,即从一点出发,不重复地走过所有的结点,最后又回到原出发点。汉密尔顿回路问题与欧拉回路问题看上去十分相似,然而又是完全不同的两个问题。汉密尔顿回路问题是访问每个结点一次,而欧拉回路问题是访问每条边一次。对图G是否存在,欧拉回路前面已给出充分必要条件,而对图G是否存在,汉密尔顿回路至今仍未找到满足该问题的充分必要条件。

2. 梵天塔问题

相传印度教的天神梵天在创造地球时,建了一座神庙。神庙里竖有3根宝石柱子,柱子由一个铜座支撑。梵天将64个直径大小不一的金盘子,按照从大到小的顺序依次套放在第1根柱子上,形成一座金塔,如图0-3所示,即所谓的梵天塔,又称汉诺塔。天神让庙里的僧侣们将第一根柱子上的64个盘子,借助第2根柱子,全部移到第3根柱子上,即将整个塔

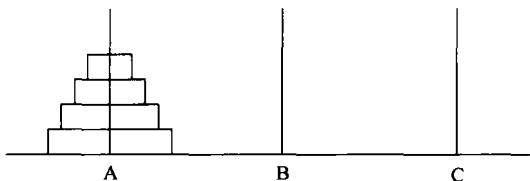


图 0-3 梵天塔问题示意图

Z 迁移。同时定下 3 条规则。

- (1) 每次只能移动一个盘子。
- (2) 盘子只能在 3 根柱子上来回移动，不能放在他处。
- (3) 在移动过程中，3 根柱子上的盘子必须始终保持大盘在下，小盘在上。

天神说，当这 64 个盘子全部移到第 3 根柱子上后，世界末日就要到了。这就是著名的梵天塔问题。

用计算机求解一个实际问题，首先要从这个实际问题中抽象出一个数学模型，然后设计一个解此数学模型的算法，最后根据算法编写程序，经过调试、编译、连接和运行，完成该问题的求解。从实际问题中抽象出一个数学模型的实质，其实就是要用数学的方法抽取其主要的本质的内容，最终实现对该问题的正确认识。

梵天塔问题是一个典型的只有用递归方法，而不能用其他方法来解决的问题。递归是计算学科中的一个重要概念，所谓递归就是将一个较大的问题，归约为一个或多个子问题的求解方法。当然要求这些子问题比原问题简单一些，且在结构上与原问题相同。

根据递归方法，我们可以将 64 个盘子的梵天塔问题，转化为求解 63 个盘子的梵天塔问题。如果 63 个盘子的梵天塔问题能够解决，则可以先将 63 个盘子先移动到第 2 个柱子上，再将最后一个盘子直接移动到第 3 个柱子上，最后又一次将 63 个盘子从第 2 个柱子移动到第 3 个柱子上。这样则可以解决 64 个盘子的梵天塔问题。依此类推，63 个盘子的梵天塔求解问题，可以转化为 62 个盘子的梵天塔求解问题。62 个盘子的梵天塔求解问题，又可以转化为 61 个盘子的梵天塔求解问题。直到 1 个盘子的梵天塔求解问题。再由 1 个盘子的梵天塔的求解求出 2 个盘子的梵天塔直到解出 64 个盘子的梵天塔问题。

下面用 C 语言对该问题的求解算法进行描述：

```

hanoi(int n,char left,char middle,char right)
{
    if(n==1) move(1,one,_,three);
    else
    {
        hanoi(n-1,left,right,middle);
        move(1,left,_,right);
        hanoi(n-1,middle,left,right);
    }
}

```

其中， n 表示 n 个盘子的梵天塔问题， $left$ 表示第一个柱子， $middle$ 表示第 2 个柱子， $right$ 表示第 3 个柱子。函数 $hanoi(n-1,left,right,middle)$ 表示 $n-1$ 阶梵天塔，从第 1 个柱子借助第 3 个柱子先移到第 2 个柱子上，函数 $move(1,left,_,right)$ 表示将第 1 个柱子上最后一个盘子，直接放到第 3 个柱子上。函数 $hanoi(n-1,middle,left,right)$ 表示 $n-1$ 个盘子，从第 2 个柱子借助第 1 个柱子移到第 3 个柱子上。

在以上 C 语言描述的算法基础上作适当扩充，就可以形成一个完整的程序。经过编译和连接后，计算机就可以执行这个程序，并严格地按照递归的方法将问题求解出来。

用数学的方法我们也可以这样解题：假设有 n 个盘子，移动次数是 $f(n)$ 。显然 $f(1)=1$ ； $f(2)=3$ ； $f(3)=7$ ；且 $f(k+1)=2 \times f(k) + 1$ 。此后不难证明 $f(n)=2^n-1$ 。当 $n=64$ 时，

$$f(64) = 2^{64} - 1 = 18\ 446\ 744\ 073\ 709\ 551\ 615$$

现在的问题是，当 $n=64$ 时，即有 64 个盘子时需要移动多少次盘子？要用多少时间？按照上面的算。

我们已经求出需要移动盘子的次数为 $2^{64} - 1 = 18\ 446\ 744\ 073\ 709\ 551\ 615$ 次。如果每秒移动一次，一年有 31 536 000 秒，则僧侣们一刻不停地来回搬动也需要花费大约 5 849 亿年的时间。假定计算机以每秒 1 000 万个盘子的速度进行搬迁，则需要花费大约 58 490 年的时间。

就这个例子读者可以了解到，理论上可以计算的问题，实际上并不一定能行。这属于算法复杂性方面的研究内容。

3. 哲学家共餐问题

对哲学家共餐问题可以作这样的描述。5 个哲学家围坐在一张圆桌旁，每个人的面前摆有一碗面条，碗的两旁各摆有一只筷子。

假设哲学家的生活，除了吃饭就是思考问题。（这是一种抽象，即对该问题而言，其他活动都无关紧要）而吃饭的时候，需要左手拿一只筷子，右手拿一只筷子，然后开始进餐。吃完后，又将筷子摆回原处继续思考问题。那么一个哲学家的生活进程可表示为：

- (1) 思考问题；
- (2) 饿了停止思考左手拿一只筷子如果左侧哲学家已持有它则需等待；
- (3) 右手拿一只筷子如果右侧哲学家已持有它则需等待；
- (4) 进餐；
- (5) 放右手筷子；
- (6) 放左手筷子；
- (7) 重新回到思考问题状态 1。

现在的问题是，如何协调 5 个哲学家的生活进程使得每一个哲学家最终都可以进餐。

考虑下面的两种情况。

(1) 按哲学家的活动进程，当所有的哲学家都同时拿起左手筷子时，则所有的哲学家都将拿不到右手的筷子，并处于等待状态。那么哲学家都将无法进餐最终饿死。

(2) 将哲学家的活动进程修改一下，变为当右手的筷子拿不到时，就放下左手的筷子。这种情况是不是就没有问题？不一定。因为可能在一个瞬间所有的哲学家都同时拿起左手的筷子，则自然拿不到右手的筷子，于是都同时放下左手的筷子。等一会又同时拿起左手的筷子，如此这样永远重复下去，则所有的哲学家一样都吃不到饭。

以上两个方面的问题，其实反映的是程序并发执行时进程同步的两个问题：一个是死锁 (Deadlock)，另一个是饥饿 (Starvation)。

为了提高系统的处理能力和计算机的利用率，并发程序被广泛地使用。因此必须彻底解决并发程序中的死锁和饥饿问题，于是人们将 5 个哲学家问题推广为更一般性的 n 个进程和 m 个共享资源的问题，并在研究过程中给出了解决这类问题的不少方法和工具，如 Petri 网并发程序语言等工具。

与程序并发执行时进程同步有关的经典问题，还有读写者问题 (ReaderWriter Problem)、理发师睡眠问题 (Sleeping Barber Problem)。

针对以上提到的这些计算机科学中的典型问题，在计算机科学发展的当中，人们提出了很多解决方法，我们在后续的章节中将会分别介绍。希望读者在阅读本书的时候，主要领会解决这些问题的方法，甚至可以经常跳出本书的框架，结合自己的经验，试试看能否找到更