

21世纪高等学校计算机基础教育系列教材

21 SHIJI GAODENG XUOXIAO JISUANJI JICHU JIAOYU XILIE JIAOCAI

计算机科学

概论

- 赵欢 主编
- 李仁发 主审



人民邮电出版社
POSTS & TELECOM PRESS

21 世纪高等学校计算机基础教育系列教材

计算机科学概论

赵 欢 主编
李仁发 主审

人民邮电出版社

图书在版编目 (CIP) 数据

计算机科学概论 / 赵欢主编. —北京: 人民邮电出版社, 2004.11

(21 世纪高等学校计算机基础教育系列教材)

ISBN 7-115-12807-3

I. 计... II. 赵... 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 (2004) 第 108091 号

内 容 提 要

本书依广度优先的原则, 系统、通俗易懂地概述计算机科学技术的基本概念、方法和技术, 内容涉及计算机系统结构 (包括计算机组成与结构、操作系统与网络和计算机系统)、计算机软件 (包括算法、程序设计语言和软件工程)、数据组织 (包括数据结构、文件结构和数据库结构) 和高级主题 (包括人工智能、计算理论、编译技术、通信概论、信息安全和计算机应用新领域) 等。本书主要培养大学生有关计算机科学技术的基本概念, 使他们从广度上掌握计算机科学技术的基本概念, 给学生提供线索, 为将来的发展提供更大的空间。

本书可作为大学本科非计算机专业的计算机基础课程必修或选修教材, 也可以作为大学本科计算机专业的入门教材。

21 世纪高等学校计算机基础教育系列教材

计算机科学概论

◆ 主 编 赵 欢
主 审 李仁发
责任编辑 邹文波

◆ 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>
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16
印张: 21.75
字数: 524 千字
印数: 11 001 - 17 000 册

2004 年 11 月第 1 版
2006 年 7 月北京第 3 次印刷

ISBN 7-115-12807-3/TP · 4294

定价: 38.00 元

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

前 言

半个多世纪以来, 计算机科学与技术得到了迅猛发展, 以信息获取、表示、储存、处理和控制在主要研究对象的计算机科学与技术学科已深入到人类活动的各个领域, 对人类社会的发展产生了巨大的影响。当今, 人类正步入信息时代, 作为走在时代前列的大学生, 掌握与计算机科学技术相关的基本概念、一般方法和新技术, 学习和研究计算机科学与技术并利用其为本专业服务是必备之功。

近年来, 随着中小学教学创新改革的深入, 信息技术课程已经在中小学普遍开设, 其教学内容迅速涵盖了高校开设的第一门计算机基础课程的绝大部分, 大学计算机基础课程教学改革势在必行。同时我们又注意到: 一方面, 许多有条件的中学甚至小学的信息技术课程教学手段先进、教学内容不断更新, 进入大学时, 这部分学生的计算机基本操作能力已经很强; 另一方面, 不少边远农村、山区的中学没有计算机, 也没有相应师资, 学生进入大学时才首次接触计算机。面对水平如此参差不齐的学生, 大学计算机基础课, 特别是第一门课应该讲授什么内容、该如何教? 这恐怕是所有高校近几年以及未来几年都要面临的问题。大学计算机基础课如何定位? 如何体现时代的发展? 如何给我们的大学毕业生即将来的生力军, 打下一个广博发展的厚重基础? 是我们计算机教育工作者必需思考的课题。

正是基于上述思考, 编者所在的湖南大学计算机与通信学院率先对大学计算机基础课程进行了改革, 将大学的第一门计算机课程定名为“大学信息技术基础”, 分为以下两部分组织教学。

1. 计算机操作实践部分

在专用教室上课, 采用案例驱动教学, 内容为 Windows、Word、Excel、PowerPoint 以及上网浏览搜索和收发电子邮件等基本操作。

新生入学时进行摸底考试, 通过者可以申请免修。

2. 计算机科学概论部分

在多媒体教室授课, 以通俗易懂的方式系统介绍计算机科学技术的几大知识体系, 如计算机系统结构、计算机软件、数据组织和高级主题等, 培养大学生有关计算科学技术的知识理念, 使之从广度上掌握计算机科学技术的基本概念, 给学生提供线索, 为将来学生的发展提供更大的空间。这样做, 从本质上提升了大学计算机基础课程品位, 与中学信息技术课程加以真正区别。同政治理论、大学英语和大学数学一样, 计算机科学概论成为湖南大学所有专业学生必须修读的课程。

选派具有博士学位、教学效果优良的专业课教师组成教师梯队。

本教材正是这项教学改革的成果。

全书共分 16 章。第 0 章和附录由李仁发编写, 第 1 章由彭蔓蔓编写, 第 2 章由肖德贵编写, 第 3 章由赵欢编写, 第 4 章由吴昊编写, 第 5 章由李丽娟编写, 第 6 章由洪跃山编写, 第 7 章由骆嘉伟编写, 第 8 章由陈治平编写, 第 9 章由李智勇编写, 第 10 章由朱宁波编写,

第 11 章和第 12 章由李肯立编写, 第 13 章由黄生叶编写, 第 14 章由徐向阳编写, 第 15 章由杨高波编写。全书由赵欢统稿, 由湖南大学计算机与通信学院院长李仁发教授主审。

在组织教学时, 教师可根据学校所开设课时多少和学生的专业、层次, 适当删减某些章节, 如第 3 章及高级主题部分可作为选讲。

绪论在教学时可以不专门安排课堂学时, 由教师引导学生自行阅读, 建议学生先在课前阅读, 课程结束后再次阅读, 会有更深体会。

大学计算机基础课程改革百花齐放、百家争鸣, 各有所长, 同时由于作者水平有限, 书中一定存在许多不足之处, 敬请不吝指正。

编 者

2004 年 8 月于湖南长沙岳麓山

目 录

第 0 章 绪论	1
0.1 计算	1
0.2 算法	3
0.3 计算机	5
0.4 计算机学科	7
0.5 小结	7

第一部分 计算机系统结构

第 1 章 计算机组成与结构	11
1.1 计算机的发展历史	11
1.1.1 第一代计算机	11
1.1.2 第二代计算机	12
1.1.3 第三代计算机	12
1.1.4 第四代计算机	13
1.1.5 第五代计算机	13
1.2 数据的表示与基本运算	14
1.2.1 进位计数制及其转换	14
1.2.2 数值数据的表示	16
1.2.3 非数值数据的表示	24
1.2.4 数据信息的校验	27
1.2.5 数据的基本运算	28
1.3 计算机系统的基本组成	33
1.3.1 计算机系统的硬件组成	34
1.3.2 计算机系统的软件组成	39
1.3.3 计算机系统的工作过程	41
1.4 计算机的特点与性能指标	43
1.4.1 计算机的特点	43
1.4.2 计算机的性能指标	44
1.5 小结	46
习题	48
第 2 章 操作系统与网络	52
2.1 操作系统的定义	52
2.2 操作系统的演化	53
2.3 操作系统的组成	54
2.3.1 进程管理器	54

2.3.2	存储管理器	56
2.3.3	文件管理器	57
2.3.4	设备管理器	58
2.3.5	用户界面	58
2.4	主流操作系统	58
2.5	计算机网络	59
2.5.1	计算机网络基础	60
2.5.2	互联网	63
2.6	网络模型和协议	65
2.6.1	OSI 模型	66
2.6.2	TCP/IP 协议族	67
2.7	小结	69
	习题	70
第 3 章	计算机系统	72
3.1	计算机系统的分类	72
3.1.1	Flynn 分类法	72
3.1.2	冯式分类法	74
3.2	计算机系统设计技术	76
3.2.1	计算机系统设计的定量原理	76
3.2.2	计算机系统设计者的主要任务	79
3.2.3	计算机系统设计的主要方法	80
3.3	微机系统	81
3.3.1	微机系统的三个层次	81
3.3.2	Intel 微处理器	82
3.4	微机的主机系统	84
3.4.1	微机主板	84
3.4.2	微机总线	86
3.5	PC 的分类	88
3.5.1	台式机	88
3.5.2	便携机 (笔记本 PC)	88
3.5.3	个人数字助理 (PDA)	89
3.5.4	手持计算机 (HPC)	90
3.6	小结	91
	习题	91
第二部分 计算机软件		
第 4 章	算法	95
4.1	算法的概念	95
4.1.1	概述	95

4.1.2	算法定义	96
4.1.3	算法的性质	96
4.1.4	算法的结构	97
4.2	算法的表示	99
4.2.1	原语	99
4.2.2	流程图	99
4.2.3	伪码	101
4.3	基本算法	105
4.3.1	求和	105
4.3.2	求积	105
4.3.3	求最大和最小	106
4.3.4	排序	106
4.3.5	查找	107
4.4	算法效率	108
4.4.1	算法的规模	109
4.4.2	时间复杂度	109
4.4.3	空间复杂度	110
4.5	小结	111
	习题	111
第 5 章	程序设计语言	113
5.1	程序设计语言的演化	113
5.1.1	机器语言	114
5.1.2	符号语言	114
5.1.3	高级语言	115
5.2	程序设计语言的类型	117
5.2.1	过程化语言	117
5.2.2	函数式语言	118
5.2.3	逻辑式语言	118
5.2.4	面向对象语言	119
5.2.5	专用语言	120
5.3	程序设计的基本概念	121
5.3.1	标识符	122
5.3.2	变量与数据类型	122
5.3.3	常量和文字	123
5.3.4	表达式和赋值语句	124
5.3.5	控制语句	126
5.3.6	注释	128
5.4	程序单元	129
5.4.1	过程	129

5.4.2	参数	130
5.4.3	函数	130
5.4.4	输入与输出	131
5.5	程序设计语言的执行	132
5.5.1	程序翻译	132
5.5.2	链接程序	133
5.5.3	集成开发环境	134
5.6	高级话题	135
5.6.1	面向对象程序设计	135
5.6.2	程序设计语言的发展趋势	137
5.7	小结	138
	习题	139
第 6 章	软件工程	141
6.1	软件工程概论	141
6.2	软件生命周期	141
6.2.1	软件生命周期思想	142
6.2.2	软件生命周期阶段	142
6.2.3	软件生命周期模型	144
6.3	结构化方法	145
6.3.1	结构化分析	145
6.3.2	结构化设计	146
6.3.3	结构化编程	147
6.4	面向对象方法	147
6.4.1	UML 语言	147
6.4.2	面向对象分析 (OOA)	147
6.4.3	面向对象设计 (OOD)	149
6.4.4	面向对象编程 (OOP)	149
6.4.5	设计模式	149
6.5	软件项目管理	150
6.5.1	CMM 软件规范	150
6.5.2	软件需求管理	151
6.5.3	软件计划管理	151
6.5.4	软件质量管理	151
6.5.5	软件配置管理	152
6.6	文档	152
6.6.1	系统文档	152
6.6.2	用户文档	153
6.7	小结	153
	习题	153

第三部分 数据组织

第 7 章 数据结构	157
7.1 基本概念和术语	157
7.1.1 什么是数据结构	157
7.1.2 数据的逻辑结构	159
7.1.3 数据的存储结构	159
7.1.4 数据的运算	160
7.2 数组	160
7.2.1 一维数组的定义	160
7.2.2 二维数组	161
7.2.3 二维数组的存储方式	161
7.3 记录	162
7.3.1 记录的概念	162
7.3.2 访问记录	163
7.4 链表	163
7.4.1 链表的定义及特点	163
7.4.2 链表的结点结构	163
7.4.3 单链表运算	164
7.4.4 循环链表	166
7.4.5 双(向)链表	166
7.5 堆栈	166
7.5.1 堆栈的定义及基本概念	166
7.5.2 栈的实现	167
7.5.3 栈的操作	167
7.5.4 栈的应用	168
7.6 队列	169
7.6.1 队列的定义	169
7.6.2 队列的实现	169
7.6.3 队列的基本操作	170
7.6.4 队列的应用	171
7.7 树	171
7.7.1 树的概念	172
7.7.2 树结构的基本术语	172
7.7.3 二叉树的基本概念	173
7.7.4 特殊的二叉树	173
7.7.5 二叉树的实现	174
7.7.6 二叉树的遍历	175
7.7.7 二叉树的应用	176

7.8	图	176
7.8.1	图的定义和术语	177
7.8.2	图的实现	178
7.8.3	图的基本操作	179
7.8.4	图的应用	180
7.9	小结	181
	习题	182
第 8 章	文件结构	184
8.1	文件系统的任务	184
8.2	文件	184
8.2.1	文件命名	184
8.2.2	文件类型	185
8.2.3	文件属性	186
8.2.4	文件存取	187
8.2.5	编程概念	187
8.3	文件结构	189
8.3.1	顺序文件	189
8.3.2	索引文件	191
8.3.3	散列文件	192
8.3.4	编程概念	195
8.4	小结	195
	习题	196
第 9 章	数据库结构	198
9.1	概论	198
9.1.1	引言	198
9.1.2	数据库管理系统 (DBMS)	200
9.1.3	数据库的三层抽象模式	200
9.2	数据库模型	201
9.2.1	实体关系	201
9.2.2	数据库模型	202
9.3	关系数据库与 SQL	205
9.3.1	关系数据库的设计	205
9.3.2	关系的操作	207
9.3.3	结构化查询语言 (SQL)	209
9.4	数据库访问授权	211
9.4.1	数据库的权限	212
9.4.2	授权传递与回收	212
9.4.3	权限管理的 SQL 语句	212
9.5	几个高级话题	214

9.5.1 并发控制与完整性协议	214
9.5.2 数据库系统的体系结构	216
9.5.3 数据库的新应用	217
9.6 小结	220
习题	221

第四部分 高级主题

第 10 章 人工智能	225
10.1 人工智能概述	225
10.1.1 什么是人工智能	225
10.1.2 人工智能的研究途径	226
10.1.3 人工智能技术的应用	228
10.2 推理与搜索	231
10.2.1 产生式系统	232
10.2.2 搜索树	234
10.2.3 盲目搜索与启发式搜索	235
10.3 人工神经网络	239
10.3.1 基本概念	239
10.3.2 一个特殊的应用	241
10.3.3 联想记忆	244
10.4 遗传算法	246
10.4.1 遗传算法概要	246
10.4.2 遗传算法的运算过程	248
10.5 人工智能的发展趋势	249
10.6 小结	250
习题	250
第 11 章 计算理论	254
11.1 问题与计算	254
11.1.1 可计算理论	254
11.1.2 计算复杂性理论	255
11.1.3 计算理论中的数学基础	255
11.2 图灵机	259
11.2.1 图灵机的基本原理	259
11.2.2 图灵机的形式定义	260
11.2.3 图灵机的变种	261
11.3 不可计算问题	261
11.3.1 停机问题	262
11.3.2 停机问题的证明	262
11.4 问题的求解难度	263

11.4.1	算法的复杂度	263
11.4.2	P 问题和 NP 问题	264
11.5	密钥系统	265
11.5.1	密钥系统概述	265
11.5.2	背包加密算法	266
11.6	小结	267
	习题	268
第 12 章	编译技术	269
12.1	编译简介	269
12.1.1	编译器	269
12.1.2	编译器的作用	270
12.2	与编译器相关的程序	271
12.2.1	解释程序	271
12.2.2	汇编程序与连接程序	271
12.2.3	装入程序	272
12.2.4	预处理器与编辑器	272
12.2.5	调试程序与描述器	272
12.2.6	项目管理程序	273
12.3	翻译步骤	273
12.3.1	扫描程序	274
12.3.2	语法分析程序	274
12.3.3	语义分析程序	275
12.3.4	源代码优化程序	275
12.3.5	代码生成器	276
12.3.6	目标代码优化程序	277
12.4	编译器中的主要数据结构	277
12.4.1	记号	277
12.4.2	语法树	278
12.4.3	符号表	278
12.4.4	常数表	278
12.4.5	中间代码和临时文件	278
12.5	编译器结构中的其他问题	279
12.5.1	分析和综合	279
12.5.2	前端和后端	279
12.5.3	遍	280
12.5.4	语言定义和编译器	280
12.5.5	编译器的选项和界面	281
12.5.6	出错处理	281
12.6	自举与移植	281

12.7 TINY 样本语言与编译器	283
12.7.1 TINY 语言	284
12.7.2 TINY 编译器	284
12.7.3 TM 机	285
12.8 小结	286
习题	287
第 13 章 通信概论	289
13.1 计算机与通信的关系	289
13.1.1 计算机需要通信	289
13.1.2 通信需要计算机	289
13.1.3 通信发展简史	290
13.2 信息的度量与传输	291
13.2.1 信息量	291
13.2.2 信息的传输速率	291
13.3 通信系统的组成及分类	292
13.3.1 通信系统的组成	292
13.3.2 通信系统的分类	292
13.3.3 通信系统的质量指标	293
13.4 信号的传输及处理	293
13.4.1 模拟信号和数字信号的相互转换	293
13.4.2 调制与解调	294
13.4.3 复用	294
13.5 编码与解码技术	295
13.5.1 检错与纠错	295
13.5.2 压缩与解压缩	295
13.5.3 加密与解密	297
13.6 小结	298
习题	298
第 14 章 信息安全	299
14.1 信息安全的基本概念	299
14.2 密码技术	300
14.2.1 基本概念	300
14.2.2 密码技术的发展阶段	300
14.2.3 古典密码算法	301
14.2.4 对称密码算法	301
14.2.5 非对称密码算法	303
14.3 数字签名技术	304
14.3.1 单向散列函数	304
14.3.2 数字签名	305

14.3.3 数字签名的应用	306
14.4 防火墙技术	306
14.4.1 防火墙的概念	306
14.4.2 防火墙的作用	306
14.4.3 防火墙的基本类型	307
14.4.4 防火墙的优缺点	309
14.5 计算机病毒	309
14.5.1 计算机病毒的概念	310
14.5.2 计算机病毒分类	311
14.5.3 计算机病毒的检测与防治	311
14.6 小结	312
习题	313
第 15 章 计算机应用新领域	314
15.1 计算机的发展历程	314
15.2 计算机及其应用现状	316
15.2.1 计算机及其应用现状	316
15.2.2 计算机应用面临的问题	316
15.3 计算机的发展趋势与未来的计算机	319
15.4 计算机应用的新领域	321
15.4.1 多媒体娱乐	321
15.4.2 3D 浸入式视频会议	322
15.4.3 网格计算	323
15.5 小结	323
习题	324
附录	326
计算机科学与技术学科知识体系简介	327
参考文献	332

第 0 章

绪论

本书是为大学低年级学生编写的。目的在于按照广度优先的原则，向学生系统地介绍计算机学科的基本内容，使学生对计算机、计算机学科有一个正确的、基本的了解。

计算机科学作为一门独立学科的历史并不长。20 世纪 70 年代，计算机技术得到迅猛发展，并逐步渗透到各个领域，但计算机科学是不是一门独立的学科，却一直存有争议。经过美国计算机协会（The Association for Computing Machinery, ACM）和美国电气和电子工程师协会下属的计算机学会（The Computer Society of the Institute for Electrical and Electronic Engineers, IEEE-CS）近四年的努力，完成了题为《计算作为一门学科》（Computing as a Discipline）的报告，报告的主要内容发表在 1989 年 1 月的 *Communications of the Acm* 杂志上。报告基本上完成了计算机科学作为一门独立学科的存在性证明，并从教育的角度提出了较为科学与完整的课程体系。

计算机科学，用一句通俗的话来说，就是一门研究与计算机有关问题的科学，因此，它的内涵和外延都十分广泛和复杂。但广义地说，应主要包括计算（Computing）和计算机（Computer）两大部分，而计算和计算机是不能混为一谈的。

0.1 计 算

“计算”是一个数学概念。按一般的理解，数的加减乘除及其混合运算是计算，方程的求解是计算，函数的微分和积分是计算，定理的证明与推导也是计算。抽象地理解，我们可以这样认为：所谓计算，就是从一个符号串 f 变换成另一个符号串 g 的过程。

为了更好地理解什么是计算，我们来看以下几个例子。

例 0-1. 当将符号串 $12+4$ 变换成 16 ，这就是一个加法计算。

例 0-2. 设符号串 $f=x^2$ ，符号串 $g=2x$ ，则是从 f 到 g 的微分计算。

例 0-3. 令 f 表示一组公理及其推导规则，令 g 是一个定理，则从 f 到 g 的一系列变换，就是定理 g 的证明。这是证明型计算。

例 0-4. 如果 f 代表一个英文句子, g 为语义相同的中文句子, 那么从 f 到 g , 就是把英文句子翻译成中文句子。这是一类符号计算。

上述 4 个例子, 我们都称为计算, 因为它们有一个共同点: 都是从已知符号或符号串开始, 按照某种规则, 一步一步地改变符号或符号串, 经过有限的步骤, 最后得到一个我们所期望的符号或符号串这样一个变换过程。

进一步分析, 我们可以把计算分类。计算主要有两大类: 一类是数值计算, 另一类是符号推导。数值计算是一类包括函数的加减乘除、幂运算、开方运算和方程的求解在内的计算。符号推导是一类包括代数及各种函数恒等式、不等式证明和几何命题证明等在内的计算。两类计算形式上不同, 但本质上却是等价的, 计算的本质相同, 相互间可以转化。

根据以上分析, 我们试图给计算下一个定义。

定义 0.1: 计算是从一个已知的符号串开始, 按照一定的规则改变符号串, 经过有限步骤, 最后得到一个满足预先规定的符号串的变换过程。

计算的理论体系是复杂的, 在这里, 我们仅做一点常识性的简单介绍。

哥德尔 (K.Godel, 1906~1978) 在 1931 年提出了原始递归函数的概念。所谓原始递归函数, 就是由初始函数出发, 经过有限次的演化和递归而得到的函数。受埃尔布朗 (J.Herbrand, 1908~1931) 一封信的暗示, 哥德尔于 1934 年引进了一般递归函数的概念, 后经过克林 (S.C.Kleene, 1909~1994) 的改进与证明, 得到如下目前普遍采用的递归函数定义。

定义 0.2: 一般递归函数是指由初始函数出发, 经过有限次使用代入, 原始递归式和造逆函数的算子或求根算子而构造成的有定义的函数。

递归函数是一种计算模型。所谓递归, 即反复进行相同性质的运算, 每一次运算处理的问题均为前一次所分出来的子问题, 直到问题无需再简化为止。当有一已知的答案, 然后再逐层返回, 直到最上层递归调用开始的地方为止, 此即为欲求的答案。

从程序设计的角度来看, 构成递归需具备以下条件。

- (1) 递归程序必须具备停止条件, 否则会造成无穷递归。
- (2) 递归过程中传递变量必须不断递减, 最少每次需向下减 1, 将问题的困难度逐渐降低, 并满足停止条件的时间有限。
- (3) 任何一次递归过程, 执行上只有两种可能: 继续向下递归或直接传值下去, 一直回到递归开始处, 程序才会结束。

除一般递归函数之外, 还有 λ 演算、图灵机和波斯特 (E.L.Post, 1897~1954) 系统。4 种模型完全从不同的角度研究计算过程或证明过程。形式上相互间有很大的区别, 但本质上却完全等价。丘奇 (A.Church, 1912~1955) 和克林分别证明了 λ 可计算函数就是一般递归函数, 从而论证了两类可计算函数是等价的, 一致的。丘奇于 1936 年发表了著名的丘奇论点: **每个能行的可计算的函数, 都是一般递归函数。**

与此同时, 图灵 (A.M.Turing, 1912~1954) 提出了另一类计算模型——图灵机, 并提出了著名的图灵论点:

能行的可计算函数, 都是用图灵机可计算的函数。

一年以后, 图灵进一步证明了图灵机可计算函数与 λ 可计算函数是一致的, 当然也就和一般递归函数是一致的, 等价的。于是, 形成了著名的丘奇—图灵论点:

凡是可计算的函数都是一般递归函数, 也就是图灵机可计算函数。