



国家精品课程教材

高等学校规划教材

# 大学计算机基础

/// (第2版)

吴宁 主编      © 崔舒宁 陈文革 李威威 编著  
马博琴 主审



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

国家精品课程教材  
高等学校规划教材

---

# 大学计算机基础

---

(第2版)

主编 吴 宁

编著 崔舒宁 陈文革 李威威

主审 冯博琴

电子工业出版社

Publishing House of Electronics Industry

## 内 容 简 介

本书是国家精品课程“大学计算机基础”的主教材,是在《大学计算机基础》的基础上修订而成的,此次再版,融入了两年教学实践的新体会,同时,对部分内容进行了调整和修改。

本书以“计算思维能力”培养为主线,强调“计算机基本工作原理”的理解和“问题求解思路”的建立。在组织架构上主要分为四个部分,共8章,主要内容包括计算与可计算性理论简述(引言部分)、计算机基础知识、微型计算机系统、计算机网络及应用、Visual Basic 程序设计、数据结构、算法分析与设计和综合案例。

本书各章均在起始处给出了该章的引言及教学目的,以供读者学习时参考。同时,还配有大量图示和例题,以便于对内容的理解。为方便教学,本书还免费提供电子课件,可以登录华信教育资源网([www.hxedu.com.cn](http://www.hxedu.com.cn))免费注册下载。

本书可作为普通高等学校非计算机专业“大学计算机基础”课程的教材,适用学时为48~64学时。目录中带有“\*”的章节为可选内容,可根据具体情况选择讲授或作为课外开放性学习使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

## 图书在版编目(CIP)数据

大学计算机基础 / 吴宁主编; 崔舒宁, 陈文革, 李威威编著. —2版. —北京: 电子工业出版社, 2013.8  
高等学校规划教材

ISBN 978-7-121-15529-1

I. ①大… II. ①吴… ②崔… ③陈… ④李… III. ①电子计算机—高等学校—教材 IV. ①TP3

中国版本图书馆 CIP 数据核字 (2013) 第 169687 号

策划编辑: 索蓉霞

责任编辑: 郝黎明

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1 092 1/16 印张: 17.75 字数: 454.4 千字

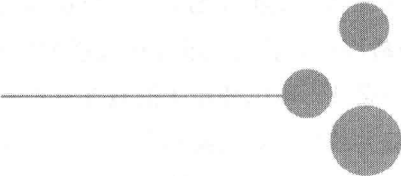
印 次: 2013 年 8 月第 1 次印刷

定 价: 35.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。



# 前言

《大学计算机基础》第1版自2011年出版以来,已在我校使用两个学年。本书是在2011年基础上修订而成的。此次再版,仍保持了第1版教材的“以计算思维能力培养为主线,强调计算机基本工作原理理解和问题求解思路建立”这一宗旨,但融入了近两年来的教学体会及新的教学研究成果,对第1版中的部分内容进行了精简,适当加强了可计算性理论及操作系统的介绍。主要修订内容如下:

(1) 按照以“计算思维”为切入点的培养需求,加入了计算与可计算性理论的描述,以帮助读者进一步了解哪类问题是计算机可计算的,哪些是不可计算的。

(2) 加强了操作系统原理描述,特别是进程调度及存储器管理等内容。目的是使读者能更深入地理解多任务并发执行的原理,以及计算机中存储器系统的构成和管理模式。

(3) 对原计算机网络部分的内容进行了修订。更新为:以互联网为主线介绍计算机网络的基本概念、工作原理和应用。同时,在有关网络安全技术的介绍上,更加具有条理性和逻辑性,不仅包括了网络安全技术原理,也包括了网络安全技术的实际应用。

(4) 为了进一步提升学习者“利用计算机求解问题的能力”,将数据结构与算法设计分为两章,对内容进行了充实,新增了一些设计示例,以帮助读者理解相关内容。

本次再版,既考虑了目前全国高等学校入校新生在计算机基础知识方面的一般现状,更考虑了创新型人才必须具备利用计算机求解问题的能力这一需求。力求使读者通过本书的学习,能够在了解计算机基础知识的基础上,较为深入地理解微型计算机的基本工作原理,能够初步建立起利用计算机解决问题的思路、掌握求解问题的一般方法,并了解利用计算机解决问题的一般过程,以及了解哪类问题是计算机不可解决的。

全书在组织架构上主要分为四个部分,共8章。

四部分内容:一是计算与可计算性理论;二是计算机中的信息表示;三是微型计算机系统组成和基本工作原理;四是算法和数据结构设计和实现。

第1章为引论,主要介绍了计算与可计算性理论、计算工具的发展、计算机问题求解过程、当前计算机科学研究的前沿技术等。第2章~第4章为计算机基础知识、系统组成、操作系统和网络技术等系统平台知识。第5章~第7章保持了第1版的主要内容,以“建立利

用计算机解决问题的思路和方法”为宗旨，主要介绍利用 VB 程序语言进行数据结构和简单算法的设计及算法描述方法等。第 8 章仍然是一些综合案例设计。

本书配备同步修订的实验指导书大学计算机基础实验教材(Windows7+Office 2010) (第 2 版)，新的实验指导教程也随着技术的发展及教学研究的成果做了一定的修订。为方便教学，本书还免费提供电子课件，可以登录华信教育资源网 ([www.hxedu.com.cn](http://www.hxedu.com.cn)) 注册下载。

本书由吴宁主编并统稿，参与编写的有吴宁 (第 1~3 章)、陈文革 (第 4 章)、崔舒宁 (第 5~8 章)、李威威整理了附录并负责全书的校对；程向前和贾应智二位老师提供了部分案例。本书由首届国家级教学名师冯博琴教授主审，他为本书提出了许多宝贵的意见和建议，藉此表示衷心的感谢。

虽然经两年多的教学体会，使此次修订较第 1 版在内容安排、编写风格等方面都有一些改进，但作为一门新兴的课程，“大学计算机基础”的教学内容及相应教材的编写依然是难点，加之作者水平所限，书中错误和不妥之处在所难免，恳望师生不吝指正，十分感谢。作者 E-mail: [wun@mail.xjtu.edu.cn](mailto:wun@mail.xjtu.edu.cn)。

编 者

于西安交通大学

2013 年 8 月



## 反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

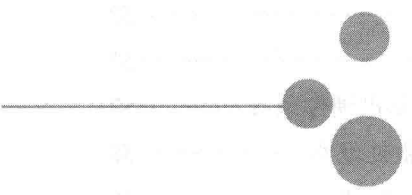
传 真：(010) 88254397

E-mail: [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036



# 目 录

|                      |    |                  |    |
|----------------------|----|------------------|----|
| 第 1 章 引论             | 1  | 1.5.1 高性能计算      | 28 |
| 引言                   | 1  | 1.5.2 普适计算       | 29 |
| 教学目的                 | 1  | 1.5.3 云计算        | 30 |
| 1.1 计算与可计算性          | 1  | 1.5.4 人工智能       | 30 |
| 1.1.1 计算与计算科学        | 2  | 1.5.5 物联网        | 32 |
| 1.1.2 可计算性理论         | 3  | 习题               | 32 |
| 1.1.3 图灵机模型          | 5  | 第 2 章 计算机基础知识    | 34 |
| *1.2 计算机的发展历程        | 7  | 引言               | 34 |
| 1.2.1 电子计算机的诞生和发展    | 7  | 教学目的             | 34 |
| 1.2.2 微型计算机的发展       | 9  | 2.1 计算机系统        | 34 |
| 1.2.3 未来计算机的发展       | 11 | 2.1.1 计算机系统构成    | 34 |
| 1.3 计算机中的信息表示        | 12 | 2.1.2 微型计算机主机板   | 36 |
| 1.3.1 信息             | 12 | 2.1.3 计算机的主要性能指标 | 39 |
| 1.3.2 数值信息表示         | 13 | 2.2 计算机中的数制      | 40 |
| 1.3.3 文字信息表示         | 15 | 2.2.1 数的编码单位     | 40 |
| 1.3.4 声音与图像信息表示      | 16 | 2.2.2 计算机中的常用计数制 | 41 |
| *1.3.5 计算机中信息处理的一般过程 | 19 | 2.2.3 各种数制之间的转换  | 43 |
| 1.4 基于计算机的问题求解       | 22 | 2.3 二进制数的表示和运算   | 45 |
| 1.4.1 需求分析与模型建立      | 23 | 2.3.1 二进制数的表示    | 45 |
| 1.4.2 模块设计           | 24 | 2.3.2 二进制数的算术运算  | 47 |
| 1.4.3 程序编码与调试        | 25 | 2.3.3 机器数的表示和运算  | 48 |
| 1.4.4 系统测试           | 27 | 2.4 逻辑运算与逻辑门     | 51 |
| *1.5 计算机科学研究的前沿技术    | 28 | 2.4.1 逻辑运算       | 52 |
|                      |    | 2.4.2 基本逻辑门      | 53 |

|                           |    |                                    |     |
|---------------------------|----|------------------------------------|-----|
| 习题 .....                  | 55 | 4.1.2 计算机网络应用模式 .....              | 105 |
| <b>第3章 微型计算机系统</b> .....  | 57 | 4.1.3 计算机网络的体系结构和协议 .....          | 106 |
| 引言 .....                  | 57 | 4.2 互联网 (Internet) .....           | 110 |
| 教学目的 .....                | 57 | 4.2.1 互联网基础 .....                  | 110 |
| 3.1 微型计算机硬件系统 .....       | 57 | 4.2.2 互联网应用 .....                  | 124 |
| 3.1.1 微处理器 .....          | 58 | 4.3 网络安全 .....                     | 130 |
| 3.1.2 存储器 .....           | 59 | 4.3.1 网络安全的基本概念 .....              | 130 |
| 3.1.3 总线 .....            | 62 | *4.3.2 信息安全技术 .....                | 132 |
| 3.1.4 输入/输出接口 .....       | 64 | *4.3.3 网络安全防护 .....                | 137 |
| 3.2 冯·诺依曼计算机 .....        | 66 | 习题 .....                           | 140 |
| 3.2.1 程序和指令 .....         | 67 | <b>第5章 Visual Basic 程序设计</b> ..... | 143 |
| 3.2.2 冯·诺依曼计算机基本结构 .....  | 68 | 引言 .....                           | 143 |
| 3.3 微型计算机的基本工作原理 .....    | 68 | 教学目的 .....                         | 143 |
| 3.3.1 指令的执行过程 .....       | 68 | 5.1 程序设计基础 .....                   | 143 |
| 3.3.2 微型计算机的一般工作过程 .....  | 71 | 5.1.1 什么是程序设计 .....                | 143 |
| 3.3.3 图灵机与计算机 .....       | 73 | 5.1.2 程序设计语言 .....                 | 144 |
| 3.4 非冯·诺依曼计算机 .....       | 77 | 5.1.3 程序的编译 .....                  | 146 |
| 3.4.1 冯·诺依曼计算机的局限性 .....  | 77 | 5.2 变量及数据类型 .....                  | 146 |
| *3.4.2 数据流计算机结构 .....     | 78 | 5.3 运算符及表达式 .....                  | 148 |
| 3.4.3 哈佛结构 .....          | 80 | 5.3.1 赋值运算符 .....                  | 148 |
| 3.5 操作系统 .....            | 81 | 5.3.2 算术运算符 .....                  | 148 |
| 3.5.1 操作系统概述 .....        | 81 | 5.3.3 关系运算符 .....                  | 149 |
| 3.5.2 进程管理 .....          | 83 | 5.3.4 逻辑运算符 .....                  | 150 |
| 3.5.3 存储器管理 .....         | 89 | 5.3.5 表达式 .....                    | 150 |
| 3.5.4 文件管理 .....          | 92 | 5.4 控制语句 .....                     | 151 |
| *3.5.5 其他功能 .....         | 95 | 5.4.1 程序的三种基本结构 .....              | 151 |
| 习题 .....                  | 97 | 5.4.2 条件分支语句 .....                 | 153 |
| <b>第4章 计算机网络及应用</b> ..... | 98 | 5.4.3 循环语句 .....                   | 154 |
| 引言 .....                  | 98 | 5.5 数组 .....                       | 156 |
| 教学目的 .....                | 98 | 5.6 子程序过程与函数过程 .....               | 158 |
| 4.1 计算机网络基础 .....         | 98 | 5.6.1 过程 .....                     | 158 |
| 4.1.1 计算机网络概述 .....       | 98 |                                    |     |



|                                             |     |                            |     |
|---------------------------------------------|-----|----------------------------|-----|
| 5.6.2 调用 Sub 过程.....                        | 159 | 6.1.2 数据结构.....            | 182 |
| 5.6.3 Function 过程.....                      | 159 | 6.2 线性表.....               | 184 |
| 5.6.4 Function 过程的调用<br>.....               | 160 | 6.2.1 线性表的逻辑结构及运算<br>..... | 184 |
| 5.6.5 参数传递.....                             | 160 | 6.2.2 线性表的存储结构.....        | 185 |
| 5.6.6 值变量和引用变量与参数<br>传递.....                | 161 | 6.2.3 List 类.....          | 191 |
| 5.6.7 Sub Main.....                         | 162 | 6.2.4 LinkedList 类.....    | 192 |
| 5.6.8 变量的作用范围.....                          | 162 | 6.3 栈和队列.....              | 194 |
| *5.6.9 递归调用.....                            | 163 | 6.3.1 栈.....               | 194 |
| 5.7 对象和类.....                               | 164 | 6.3.2 Stack 类.....         | 197 |
| 5.7.1 对象.....                               | 164 | 6.3.3 队列.....              | 199 |
| 5.7.2 类.....                                | 164 | 6.3.4 Queue 类.....         | 204 |
| 5.8 控制台的输出与输入.....                          | 165 | 6.4 图和树.....               | 206 |
| 5.8.1 控制台的输出.....                           | 165 | 6.4.1 图的基本概念.....          | 206 |
| 5.8.2 控制台的输入.....                           | 167 | 6.4.2 带权图和最短路径.....        | 207 |
| 5.9 使用 Visual Studio 2008.....              | 169 | 6.4.3 树的基本概念.....          | 210 |
| 5.9.1 控制台应用程序的创建与<br>运行.....                | 169 | 6.4.4 二叉树.....             | 212 |
| 5.9.2 Visual studio 2008 集成<br>环境.....      | 172 | *6.4.5 树的遍历.....           | 212 |
| 5.10 范例程序阅读.....                            | 174 | 习题.....                    | 213 |
| *5.11 关于 Visual Basic 2008 其他<br>应该知道的..... | 177 | 第 7 章 算法分析与设计.....         | 214 |
| 5.11.1 Visual Basic 的发展历程<br>.....          | 177 | 引言.....                    | 214 |
| 5.11.2 Visual Basic 2008 的解决<br>方案.....     | 177 | 教学目的.....                  | 214 |
| 5.11.3 良好的编程风格.....                         | 179 | 7.1 算法的基本概念.....           | 214 |
| 习题.....                                     | 179 | 7.2 算法的描述方法.....           | 215 |
| 第 6 章 数据结构.....                             | 181 | 7.2.1 算法的自然语言描述<br>.....   | 215 |
| 引言.....                                     | 181 | 7.2.2 算法的伪代码描述.....        | 216 |
| 教学目的.....                                   | 181 | 7.2.3 算法的流程图描述.....        | 216 |
| 6.1 数据与数据结构.....                            | 181 | 7.3 算法的复杂性评价.....          | 219 |
| 6.1.1 数据.....                               | 181 | 7.3.1 算法的时间复杂度.....        | 219 |
|                                             |     | 7.3.2 算法的空间复杂度.....        | 220 |
|                                             |     | 7.4 查找算法.....              | 220 |
|                                             |     | 7.4.1 顺序查找.....            | 220 |
|                                             |     | 7.4.2 折半查找.....            | 221 |
|                                             |     | 7.5 排序算法.....              | 223 |

|                              |     |                                       |     |
|------------------------------|-----|---------------------------------------|-----|
| 7.5.1 冒泡排序.....              | 223 | 8.2.1 客户端编程.....                      | 250 |
| 7.5.2 选择排序.....              | 225 | 8.2.2 ASP 编程概述 .....                  | 252 |
| *7.5.3 快速排序.....             | 226 | 8.2.3 ASP.NET 编程简介 ...                | 253 |
| *7.6 常用算法简介.....             | 229 | 8.3 数据库编程初步.....                      | 257 |
| 7.6.1 递归与分治.....             | 229 | 8.3.1 数据库系统.....                      | 257 |
| 7.6.2 动态规划.....              | 230 | 8.3.2 使用 Visual Studio 操作<br>数据库..... | 258 |
| 7.6.3 贪心算法.....              | 233 | 8.3.3 在 Visual Basic 中访问<br>数据库.....  | 260 |
| 7.6.4 回溯法.....               | 234 |                                       |     |
| 习题 .....                     | 236 | 附录 A 常用外设及设备驱动程序.....                 | 264 |
| *第 8 章 综合案例设计.....           | 237 | 一、输入设备 .....                          | 264 |
| 引言 .....                     | 237 | 二、输出设备 .....                          | 265 |
| 教学目的 .....                   | 237 | 三、设备驱动程序.....                         | 268 |
| 8.1 Windows 环境下编程简介 .....    | 237 | 附录 B 标准 ASCII 表 .....                 | 270 |
| 8.1.1 Windows 的消息机制<br>..... | 237 | 附录 C 声音、图像信息的数字化 .....                | 271 |
| 8.1.2 常用控件.....              | 244 | 一、声音信息的数字化 .....                      | 271 |
| 8.1.3 编程实例.....              | 247 | 二、图像信息的数字化 .....                      | 273 |
| 8.2 网络编程 .....               | 249 | 参考文献 .....                            | 275 |

# 第 1 章 引 论

## 引言

1991 年,美国施乐公司 PARC 研究中心的 Mark Weiser 在 Scientific American 上发表题为“Computer for the 21th Century”的文章中,提出了“无处不在的计算 (Ubiquitous Computing)” (又译为“普适计算”)的理念,开创了计算领域的第三次浪潮<sup>[1]</sup>。无处不在的计算设备,无处不在的网络和通信,彻底改变了人类数千年的生活习惯。人们希望通过无处不在的计算,能随时随地获得自己希望的服务,且不用关心这些服务是怎样得到的。由于提供这些服务或计算的重要载体是计算机,因此,现代信息社会的每个人,都需要了解计算机,学习计算机科学。而作为未来的工程计算机人员和科学家,更需要具备利用计算机解决相关问题的能力,以及判断什么样的问题可以由计算机解决的能力。作为全书的“引论”,本章将从什么是计算开始,简要介绍可计算性理论、计算机中的信息表示、基于计算机的问题求解过程,以及当前计算机科学研究的一些前沿技术。

## 教学目的

- 了解计算与可计算性。
- 理解图灵机模型及其工作过程。
- 了解计算机的发展。
- 了解计算机中的信息表示及信息的处理过程。
- 了解基于计算机的问题求解过程。
- 了解当前计算机科学研究的前沿技术。

## 1.1 计算与可计算性

今天,计算机已深入到生活的各个角落,几乎每个人都知道计算机能做很多事,并且似乎是什么事都能做。在各类关于机器人的影视作品中,人“机”大战也成为炫目的看点。计算机真的什么都能做吗?是否真的会出现地球被机器人统治的那一天呢?

答案是否定的。20 世纪 30 年代,图灵 (Turing) 就已经证明了计算机能力的极限性,即存在计算机不能解决的问题。这类问题又称为不可计算问题。

作为学习计算机科学的入门教材,首先使读者了解“不是任何问题都是计算机可解的”这一概念是非常必要的。虽然在现代科学研究中,没有计算机是万万不能的,但计算机确实不是万能的。

[1] 计算的第一次浪潮是主机计算,人们通过字符终端共享主机计算;第二次浪潮是桌面计算时代,即个人计算机网络通信时代。

### 1.1.1 计算与计算科学

#### 1. 计算

计算的行为由来已久, 考古研究说明, 在远古时代, 古人类就有了计算问题的需要和能力。人类最初的计算工具就是人类的双手, 掰着指头数数就是最早的计算方法。一个人天生有十个指头, 因此, 十进制就成为人们最熟悉的进制计数法。

由于双手的局限性, 人类开始学习用小木棍、石子、“结绳”等方法进行计算。英文中的计算一词“Calculation”来自拉丁文中的“Calculus”, 其本意就是用于计算的小石子。在 2000 多年前古代中国人发明的算筹是世界上最早的计算工具, 而具有十进制计数法和一整套计算口诀的算盘, 则可以认为是最早的“数字计算机”, 珠算口诀就是最早的体系化的算法。

但到底什么是“计算”? 这个问题直到 20 世纪 30 年代人们才从哥德尔 (K.Godel)、丘奇 (A.Church)、艾伦·麦席森·图灵 (A.M.Turing) 等科学家的研究中弄清楚是计算的本质, 更重要的是弄清楚了什么是可计算的, 什么是不可计算的。

计算, 抽象地讲, 就是从符号串  $X$  变成另一个符号串  $Y$  的过程。例如:

(1) 从符号串  $5+8-4$  变换为  $9$ , 是进行了加减计算。

(2) 将  $X=a^2$  变换为  $Y=2a$ , 进行了微分计算。

(3) 将一段英文 (符号串  $X$ ) 翻译为中文 (符号串  $Y$ ), 也是进行了计算。

从以上这几个简单示例可以看出, 计算是按照一定的、有限的规则和步骤 (算法), 将输入转换为输出的过程。

从数学的角度, 计算主要可以分为数值计算和符号推导两大类。数值计算包括各种算术运算、方程求解等, 如上例的 (1) 和 (2); 符号推导包括各种函数的等式或不等式的证明、几何命题证明等。上例中的 (3) 可以归为符号推导。

#### 2. 计算科学

随着计算机技术的发展, 计算这个原本专门的数学概念已经泛化到了人类的整个知识领域, 并成为一种普适的科学概念。计算不再仅是数学的基础技能, 而且是整个自然科学的工具。今天, 人类的研究领域越来越广, 每个领域的研究都会涉及大量的计算。作为各门学科研究的基础, 在现代计算工具的支撑下, 计算也逐渐形成体系, 成为一门独立的学科, 即计算科学, 它是涉及数学模型构建、数值分析方法及计算机实现方法的研究领域。

数学模型 (Mathematical Model) 是用数学符号、数学公式、程序、图形等对客观问题本质属性的抽象而又简洁的刻画, 它或能解释某些客观现象, 或能预测未来的发展规律, 或能为控制某一现象的发展提供某种意义下的最优策略或较好策略。数学模型的建立首先需要对现实问题进行深入细致的观察和分析, 抽象出各种关键因素并确立它们之间的关系, 再灵活巧妙地利用各种数学知识, 表述这些关系。例如, 学生综合素质测评、股市变化分析和气象综合预测等问题, 都可以建立成一个数学模型。通过研究对象特征, 分析对象的内在规律, 确定影响变化规律的因素, 利用适当的数学工具, 构造出各因素间的关系。

这种应用相关领域知识从客观问题中抽象、提炼出数学模型的过程就称为数学建模 (Mathematical Modeling)。数学建模的核心是分析和抽象,即要从复杂的客观现实中抽取反映出其变化规律的各种因素,并建立它们之间的关系。因此,建立数学模型的目的,就是为了便于分析和研究各种客观现象的运动规律,从而确立最佳的问题解决方案。

数值分析 (Numerical Analysis) 是关于数值近似算法的研究。古巴比伦人曾利用巴比伦泥板 (数值分析的最早作品之一) 来计算近似值,而不是精确值。引入数值分析的原因是因为在许多实际问题中,常常无法求得精确值,或是无法用有理数表示结果。数值分析的目的就是在合理误差范围内,求得满足精度要求的近似结果。数值分析方法在所有工程和科学领域中得到应用。例如,数值天气预报、太空船的轨迹计算、股票市值及其变异程度分析、保险公司的精算分析等。

计算机实现方法可以描述为利用计算机求解问题的方法。在模型建立的前提下,“计算机实现”的核心就是算法设计和程序设计。

虽然计算科学不完全等同于计算机科学,但计算科学,以及其他各学科的研究都离不开计算机。因此,“利用计算机分析和解决相关问题”的能力成为每一位科学工作者应具备的基本素质。这种能力又称为计算思维 (Computational Thinking)<sup>[1]</sup> 能力。

计算机科学 (Computer Science) 是主要研究计算理论、计算机及信息处理的学科。半个多世纪来,计算机科学得到飞速发展和普及,作为现代科学体系的主要基石之一,它已逐渐超出一门单独学科的范围,演变成为一种与社会、经济、能源、材料和健康等多个领域相结合的横向型科学技术。

有关计算机科学的详细定义有多种,但不论哪种定义,都强调了算法的研究。算法描述了解决某个特定问题的确切、无歧义、有限的动作序列。例如,按照某些准则获取一个年级中综合成绩最好的学生的过程,就是一种算法。遵照菜谱一步步做出一道好菜的过程,也是一种算法。计算机科学既要研究算法分析与设计理论,也同时要考虑如何在计算机上实现算法并解决实际问题。

### 1.1.2 可计算性理论

要利用计算机解决问题,或者说完成某项任务,首先需要使计算机明确按照什么样的方法和步骤工作,即确定算法。计算机能不能完成一项任务,取决于能不能设计出完成这项任务的算法。人类很早就已开始研究算法,我国的珠算口诀、求两个正整数的最大公约数所用的辗转相除法等,都可以称为算法。

可计算性理论 (Computability Theory) 是研究计算的一般性质的数学理论。由于计算的过程就是执行算法的过程,因此,可计算理论的中心课题就是将算法这一直观概念精确化,建立计算的数学模型,研究哪些是可计算的,哪些是不可计算的,以此揭示计算的实质。由于计算与算法联系在一起,因此,可计算性理论又称为算法理论。

直观上说,求解一类问题的算法是一组规则,这组规则条数有限,每一条都是可执行的 (可操作的),并且这种操作性是绝对机械的,即不论何人何时对之进行操作,只要输入数据相同,其结果都是一样的。作为算法的一组规则,至少还应包含一条有关终止计算的

[1] “计算思维”是运用计算机科学的基础概念进行问题求解、系统设计,以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。

条目。因此,从直观上看,算法具备的特征是有限性、可执行性、机械性、确定性和终止性。

在 20 世纪以前,对“算法”、“计算”这些概念似乎并不存在什么问题,人们普遍认为所有的问题都是有算法的,至少是一切数学命题都存在算法。但是 20 世纪初,人们发现有许多问题虽已经过长期研究,但仍然找不到算法,如希尔伯特第 10 问题及半群的字的问题等。于是人们开始怀疑,是否对某些问题根本就不存在算法?即存在不可计算的问题?

于是,数学家们开始了对算法概念及可计算性的精确化研究。1934 年,哥德尔(Gödel)提出了一般递归函数的概念,并指出:凡算法可计算函数<sup>[1]</sup>都是一般递归函数,反之亦然。这一定义后来被称为埃尔布朗·哥德尔·克林定义。同年,丘奇证明了他提出的  $\lambda$  可定义函数与一般递归函数是等价的,并提出算法可计算函数等同于一般递归函数或  $\lambda$  可定义函数,即著名的“丘奇论点”。用一般递归函数虽给出了可计算函数的严格数学定义,但在具体的计算过程中,就某一步运算而言,选用什么初始函数和基本运算仍有不确定性。为消除所有的不确定性,图灵和 E·波斯特各自独立地提出了抽象计算机的概念。图灵在他的“论可计算数及其在判定问题中的应用”(on Computable Numbers, with an Application to the Entscheidungsproblem)一文中,从一个全新的角度定义了可计算函数,他全面分析了人的计算过程,把计算归结为最简单、最基本、最确定的操作动作,从而用一种简单的方法来描述那种直观上具有机械性的基本计算程序,使任何机械(能行)的程序都可以归结为这些动作。这种简单的方法以一个抽象自动机概念为基础,并已证明:这种自动机能计算的函数是可计算函数,而这种自动机不能计算的函数则是不可计算的函数。这种抽象自动机被后人称为图灵机。

1937 年,图灵在他的“可计算性与  $\lambda$  可定义性”一文中证明了图灵机可计算函数与  $\lambda$  可定义函数是等价的,从而拓宽了丘奇论点,得出:算法(能行)可计算函数等同于一般递归函数或  $\lambda$  可定义函数或图灵机可计算函数,即“丘奇—图灵论点”,相当完善地解决了可计算函数的精确定义问题,对数理逻辑的发展起了巨大的推动作用。

虽然已经证明可计算函数就是图灵机可计算函数,而图灵机可计算函数是递归可枚举函数。但这样的定义或描述对初学者来讲似乎显得有点过于高深。事实上,由于图灵机与计算机可以相互模拟(见 3.3.3 节),所以,对“可计算性”问题我们也可以通俗地描述为就是可以用计算机来解决的问题。从广义上讲,如“请为我做一个汉堡”这样的问题是无法用计算机来解决的(至少目前还不行)。计算机本身的优势在于数值计算(很多如文字识别,图像处理等非数值问题都可以通过转化为数值问题),能够用计算机解决的问题一定是“可以在确定、有限步骤内被解决的问题”,即有确定算法。像哥德巴赫猜想这样的问题就不属于“可计算问题”之列,因为计算机没有办法给出数学意义上的证明。

有关可计算性理论的深入描述因篇幅及课程性质所限,不再在此做进一步的讨论。本节讨论可计算性问题的目的,是希望读者能初步了解:[计算机不可能解决世界上所有的问题。其“不可解决性”反映在两个方面:一是不能在有限步骤内被解决;二是虽然有可能解决,但因过于复杂而不能在可接受的时间内解决](所有机械装置都存在复杂性的临界点)。关于后者见 7.3 节算法的复杂性评价。

分析某个问题的可计算性意义重大,它可以使人们不必把时间浪费在不可能解决的问

[1] 可计算函数定义:能够在抽象计算机上编出程序计算其值的函数。



题上, 而将精力用于可以解决的问题中。

### 1.1.3 图灵机模型

#### 1. 图灵机模型

图灵机 (Turing Machine) 是图灵 1936 年提出的一种抽象计算模型, 其基本思想: 用机器来模拟人用纸和笔进行数学运算的过程。

图灵将人的计算过程看作以下两个简单的动作:

(1) 在纸上写上或擦除某个符号;

(2) 将注意力从纸上的一个位置移动到另一个位置, 而人每一次的下一步动作走向依赖于人当前所关注的纸上某个位置的符号及人当前的思维状态。

为了模拟人的这种运算过程, 图灵构造出一台假想的 (抽象的) 机器 (图 1-1), 该机器由以下几个部分组成:

- 一条无限长的纸带 **Type**。纸带被划分成一个个连续的方格。每个格子上可包含一个来自有限字母表的符号, 字母表中有一个特殊的符号表示空白。纸带上的格子从左到右依次被编号为 0, 1, 2, ..., 纸带的右端可以无限延长。
- 一个读/写头 **Head** (图 1-1 中间的大盒子)。读/写头内部包含了一组固定的状态 (盒子上的方块) 和程序。该读/写头可以在纸带上左右移动, 它能读出当前所指的格子上的符号, 并能改变当前格子上的符号。
- 一套控制规则 **Table** (即程序)。**Table** 包括当前读/写头的内部状态, 输入数值, 输出数值, 下一时刻的内部状态。在每个时刻, 读/写头都从当前纸带上读入一个方格信息。根据当前机器所处的状态及读/写头所读入的格子上的符号来确定读/写头下一步的动作。同时, 改变状态寄存器的值, 令机器进入一个新的状态。
- 一个状态寄存器。它用来保存图灵机当前所处的状态。图灵机的所有可能状态的数目是有限的, 并且有一个特殊的状态, 称为停机状态。

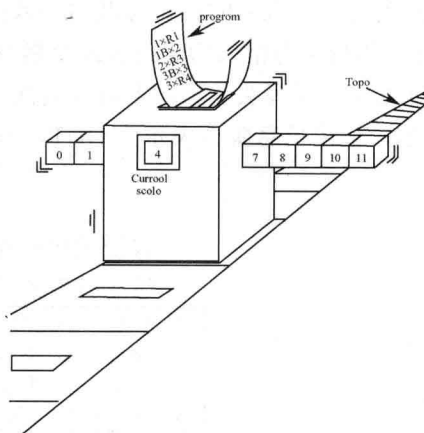


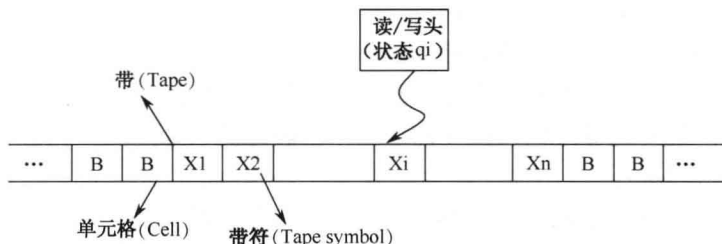
图 1-1 图灵机结构模型

图灵机根据程序的命令和内部的状态在纸带上进行移动和读/写, 它的每一部分都是有限的, 但它有一个潜在的无限长的纸带。因此, 这种机器只是一个理想的设备。图灵认为

这样的一台机器就能模拟人类所能进行的任何计算过程。

## 2. 图灵机的工作过程

将图灵机模型画成二维平面图如图 1-2 所示。首先将输入符号串（有穷长度的从输入字母表中选择的符号串）从左到右依次填在纸带的格子（带单元）上（图 1-2 中用符号  $X_i$  表示），其他带单元保持空白（即填以空白符，用符号  $B$  表示）。空格是带符号，但不是输入符号，除了输入符号和空格之外，还可能其他的带符号。若读/写头（带头）位于某个带单元之上，说明图灵机正在读/写这个单元。



图灵机的工作过程就是根据读/写头内部程序的命令及内部状态进行纸带的读/写和移动。在初始状态下，读/写头位于输入的最左边单元上（第 0 号格子）。每个图灵机都有一组变换规则，图灵机的移动是当前状态和扫描的带符号的函数，根据当前读/写头指向的带符号及其自身的状态，由规则确定读/写头是否移动及移动方向。每移动一步，图灵机将：

- 改变状态。下一状态可以是任何状态或与当前状态相同。
- 在扫描的单元中写带符号。这个带符号代替扫描的单元中任何符号。所写符号可以是任意的带符号或与当前单元的符号相同。
- 向左或向右移动带头。在本书中要求带头移动而不允许带头保持静止。这个限制并不约束图灵机的计算能力，因为包含静止带头的任何操作都可连同下一个带头移动一起被压缩成单个状态改变、写入新的带符号及向左或向右移动的操作。

**例 1-1** 假设图灵机纸带上的每个方格内为  $a$  或  $b$  或空格  $B$ ，控制规则 Table 用于表示当读入（输入）给定时（如读入  $a$ ）读/写头的移动方向（向前、向后或停止）。

将输入信息集合表示为  $IN$ ，输出信息集合表示为  $OUT$ ，且有：

$IN = \{x, y, B\}$

$OUT = \{Left, Right, Stop\}$



设图灵机纸带状态如图 1-3 所示。

给定控制规则 Table 如下：

- ① 从最左端起始；
- ② 若读入  $x$ ，右移一格；
- ③ 若读入  $y$ ，将其改为  $x$ ，再继续右移一格；
- ④ 若读入  $B$ ，停止。

依据上述规则，移动 7 次后，将纸带上

停止符  $B$  左端的符号均改为  $x$ ，并停止。

因此，图灵机的工作过程可以简单地描述为读/写头从纸带上读出一个方格中的信息，然后根据它内部的状态对程序进行查表（规则表 Table），得出一个输出动作，确定是向纸

带上写信息还是使读/写头向前或向后移动到下一个方格。同时，程序还会说明下一时刻内部状态转移到哪里。

图灵机只要根据每一时刻读/写头读到的信息和当前的内部状态，查表就可确定它下一时刻的内部状态和输出动作。只要改变规则 Table，图灵机就可以做不同的工作。如果把图灵机的内部状态解释为指令，将规则解释为程序，都用二进制编码表示，与输出和输入信息（也可用二进制码表示）同样存储在机器里，编写不同的程序就会使机器做不同的动作。这就成为电子计算机了。所以，图灵机就是一个最简单的计算机模型。

### 3. 图灵机的形式化描述

形式上，图灵机 (TM) 可以描述为一个七元组，即

$$M=(Q, \Sigma, \Gamma, \delta, q_0, B, F) \quad (1-1)$$

其中：

Q：图灵机状态的有穷集合。

$\Sigma$ ：输入符号的有穷集合，不包含空白符。

$\Gamma$ ：带符号的完整集合； $\Sigma$  是  $\Gamma$  的子集，有  $\Sigma \in \Gamma$ 。

$\delta$ ：转移函数。 $\Delta(q, X)$  的参数是状态  $q$  和带符号  $X$ 。 $\delta(q, X)$  的值在有定义时是三元组  $(p, Y, D)$ ，其中： $p$  是下一状态，属于集合  $Q$ ； $Y$  是在当前扫描的单元中写下的符号，属于  $\Gamma$  集合，代替原来单元里的符号； $D$  是方向，非  $L$  即  $R$ ，分别表示“向左”和“向右”，说明带头移动方向。

转移函数  $\delta$  是一个部分函数，换句话说，对于某些  $q$  和  $X$ ， $\delta(q, X)$  可能没有定义，如果在运行中遇到下一个操作没有定义的情况，机器将立刻停机。

$q_0$ ：初始状态，属于  $Q$ ，开始时图灵机就处于  $q_0$  状态。

$B$ ：空格符号。这个符号属于  $\Gamma$  但不属于  $\Sigma$ ，即不是输入符号。开始时，空格出现在除包含输入符号的有穷多个初始单元之外的所有单元中。

$F$ ：终结状态或接受状态的集合，是  $Q$  的子集。

今天，图灵机依然是计算理论研究的中心课题，图灵机的“停机问题”<sup>[1]</sup>更是研究许多判定问题的基础。“判定问题”是指判定无穷多个同类个别问题是否具有算法解，或者是否存在能行性的方法使得对该问题类的每一个特例都能在有限步骤内机械地判定它是否具有某种性质的问题。例如，“任给一个素数  $x$ ， $x$  是素数吗？”，这一问题就是判定问题。研究判定问题的意义：如果某问题不可判定，就意味着它是不可解的或不可计算的；相应的，若它是可计算的，则它是可判定的或可解的。由此，又归结到 1.1.2 节结尾处所述的研究可计算性理论的意义。

## \*1.2 计算机的发展历程

### 1.2.1 电子计算机的诞生和发展

在 1946 年之前，计算机的工作都是基于机械运行方式，没有进入逻辑运算领域。如果

[1] 停机问题是指是否存在一个算法，对于任意给定的图灵机都能判定任意的输入是否会导致停机？已证明图灵机的停机问题是不可判定的。停机问题的不可判定性成为了解决许多不可判定性问题的基础。