



普通高等教育“十一五”国家级规划教材



21世纪大学本科
计算机专业系列教材

石峰 计卫星 刘彩霞 编著

程序设计基础(第2版)

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

- 根据教育部“高等学校计算机科学与技术专业规范”组织编写
- 与美国 ACM 和 IEEE CS *Computing Curricula* 最新进展同步



清华大学出版社



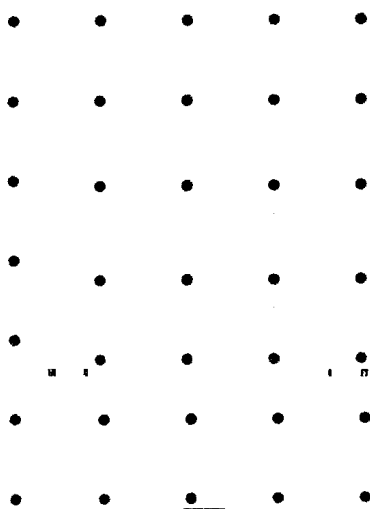
普通高等教育“十一五”国家级规划教材

21世纪大学本科计算机专业系列教材

程序设计基础

(第2版)

石峰 计卫星 刘彩霞 编著



清华大学出版社
北京

内 容 简 介

本书从程序设计的一般过程与方法出发,以 C++ 语言为媒介,重点介绍结构化程序设计与面向对象程序设计的基础知识。内容包括三个方面:程序设计概论与基础、结构化程序设计原理、面向对象程序设计原理。以分析人类解决问题的思维过程为开端,介绍计算机的问题求解过程和方法;从软件工程化的角度,讲解结构化程序设计的起源、基本原理及基本语法;从人类认知世界的过程和方法入手,总结人类认知规律,讲解面向对象技术和方法的基本原理,以及面向对象语言特有的语法。

本书融合了算法分析与设计、现代软件工程、C++ 语言、UML 建模、程序设计方法学、面向对象方法学等多方面的思想和知识,内容从易到难,覆盖面广,适用范围较宽,既适合作为高校计算机及相关专业本科生的教学用书,也可作为其他理工科专业本科生的教学参考书。对于有志于深入学习程序设计的计算机爱好者,本书也是一个好的选择。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

程序设计基础/石峰,计卫星,刘彩霞编著. —2 版. —北京:清华大学出版社,2010.4
(21 世纪大学本科计算机专业系列教材)

ISBN 978-7-302-21854-8

I. ①程… II. ①石… ②计… ③刘… III. ①C 语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 009299 号

责任编辑:张瑞庆

责任校对:李建庄

责任印制:王秀菊

出版发行:清华大学出版社

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

社 总 机:010-62770175

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

印 刷 者:北京市昌平环球印刷厂

装 订 者:三河市李旗庄少明装订厂

经 销:全国新华书店

开 本:185×260 印 张:23.75

字 数:590 千字

版 次:2010 年 4 月第 2 版

印 次:2010 年 4 月第 1 次印刷

印 数:1~3000

定 价:33.00 元

21 世纪大学本科计算机专业系列教材编委会

名誉主任：陈火旺

主 任：李晓明

副 主 任：钱德沛 焦金生

委 员：（按姓氏笔画为序）

马殿富	王志英	王晓东	宁 洪	刘 辰
孙茂松	李大友	李仲麟	吴朝晖	何炎祥
宋方敏	张大方	张长海	周兴社	侯文永
袁开榜	钱乐秋	黄国兴	蒋宗礼	曾 明
廖明宏	樊孝忠			

秘 书：张瑞庆

本书责任编辑：张长海

前言

FOREWORD

坐在计算机前,挥洒自如地驰骋于电脑世界,一行行程序在自己的一双巧手下源源不断地流出,这是一些初学编程者的美好憧憬。然而在学习了程序设计语言的一些具体语法后,真正动起手来编写程序时,却往往会觉得一筹莫展、无所适从;一些读者在遇到一种新型编程语言时,即便是已经掌握了某些编程语言,学起来也常常感到举步维艰,未能体会到已有的知识积累给学习新语言带来的便利。为什么这样?原因在于没有掌握程序设计的一般过程,没有深入了解通用程序设计语言的本质规律。

通用程序设计语言是与人类自然语言具有一定相似性的人造语言,使用通用程序设计语言进行程序设计并不能简单地等价于人们常说的“编程序”。程序设计是一个过程,通常分为三个阶段:算法分析与设计、程序编写与调试、程序测试,正像人们利用自然语言书写文章一样,与此三阶段相对应的是构思、起草与修改、审阅。不同阶段的任务是相对独立的,如果对此没有深入的了解,则很容易将它们混为一谈,在缺乏对问题深入、全面分析的情况下匆匆动手编写程序,自然就会像没有经过构思而书写出来的文章一样,基本上会是一篇流水账,不可能成为绝唱。学习程序设计的目的在于通过学习,能够从自觉地遵守程序设计的准则,到不自觉、非有意、自然而然地按照程序设计准则去从事程序设计工作。为实现这个目的,学习算法分析和设计自然成为“程序设计基础”课程的主要学习任务之一。

程序设计归根到底要使用程序设计语言将问题的解题过程描述出来,因此学习程序设计语言是本课程的另一重要任务,本书虽然以 C++ 语言为背景介绍命令式程序设计语言的相关知识,但是应该注意到,现有的通用程序设计语言的语法和风格可能迥异,但无论哪一种语言,都是以数据(类型)、操作(运算)、(操作)顺序控制和数据控制为基本内容。也就是说,万变不离其宗,不同的语言都以不同的格式对这 4 个方面提供不同的支持。读者在学习本书的具体内容时,抛却语言的具体表述格式,深入学习和理解这 4 个方面的实质,会给学习其他程序设计语言带来便利。

与许多其他同类教材不同的是,本书强调软件开发过程的重要性,为读者介绍了有关软件开发模型与软件测试的基本原理和技术。这样做的原因在于,现代软件开发多为集体合作,这种合作过程中每一个编程者所编写程序的规范性、正确性和可靠性十分重要,在学习程序设计之初就了解程序设计的规范过程,对于培养将来参与大型软件开发所需要的分工合作精神十分重要。

全书分为三个部分:第一部分由第 1 章和第 2 章组成,讲述程序设计的一些基本概念和思想,读者在学习过程中应该以了解和掌握结构化算法设计为重点;第二部分由第 3 章至第

8章组成,以C++语言中的非面向对象语法为例,讲述一般程序设计语言所具有的数据(类型)、操作、(操作)顺序控制、数据控制基本内容,同时重点介绍如何进行结构化程序设计及软件开发的一般过程,对于这部分内容的学习,读者应该抛却具体的语言,从最一般的观点(通过程序)理解算法结构与解题过程描述;第三部分由第9章至第12章组成,重点介绍面向对象程序开发技术的基本方法,包括面向对象分析与设计的基本思想和主要步骤,以及C++中的面向对象特征的语法,该部分内容的目标是使读者了解和掌握人类认知的最一般规律,以及如何通过适当的步骤,采用面向对象技术和面向对象语言的基本语法,将自身对客观事物的认识描述出来。

三位作者共同参与和完成了整书的编写及修订工作,其中石峰主要负责全书的统筹及第一部分内容的修订;刘彩霞主要负责第二部分的修订;计卫星主要负责第三部分内容的修订。

本教材的内容从简到难,覆盖面广,有些内容涉及一些较深入的知识(标题中带有*号),教师在使用本教材时,应该根据学生及教学的具体情况对内容进行适当剪裁。总之,对于本教材,学习的重点应该放在算法、程序结构的设计以及注重以程序设计语言的习惯和方式表述问题的解决方案上,而不应放在具体语言的语法细节上。

本书中的所有例题和习题中的程序都在Windows XP环境Visual C++ 6.0编译系统下运行,读者在学习时选择Visual C++ 5.0以上的版本即可。

本书是在多年教学实践基础上编写的,适合作为高校计算机及其相关专业本科生的教学用书,也可用于其他理工科专业的计算机教学参考书。对于有志于深入学习程序设计的广大计算机爱好者,本书也是一个好的选择。书中所用到的部分素材来源于教学过程中的积累和其他中外文教材、资料,由于无法在此一一列举,这里谨对这些教材和资料的作者表示衷心的感谢。对于在教材编写过程中,吉林大学张长海教授及编委会成员给予的关心和指导,以及清华大学出版社所给予的大力支持,作者在此表示最衷心的感谢!对于书中一些错误和有待商榷之处,欢迎各位专家不吝赐教。

作 者

2010年1月于北京理工大学

读者意见反馈

亲爱的读者：

感谢您一直以来对清华版计算机教材的支持和爱护。为了今后为您提供更优秀的教材，请您抽出宝贵的时间来填写下面的意见反馈表，以便我们更好地对本教材做进一步改进。同时如果您在使用本教材的过程中遇到了什么问题，或者有什么好的建议，也请您来信告诉我们。

地址：北京市海淀区双清路学研大厦 A 座 602 室 计算机与信息分社营销室 收

邮编：100084

电子邮件：jsjic@tup.tsinghua.edu.cn

电话：010-62770175-4608/4409

邮购电话：010-62786544

教材名称：程序设计基础（第 2 版）

ISBN：978-7-302-21854-8

个人资料

姓名：_____ 年龄：_____ 所在院校/专业：_____

文化程度：_____ 通信地址：_____

联系电话：_____ 电子信箱：_____

您使用本书是作为：☐指定教材 ☐选用教材 ☐辅导教材 ☐自学教材

您对本书封面设计的满意度：

☐很满意 ☐满意 ☐一般 ☐不满意 改进建议_____

您对本书印刷质量的满意度：

☐很满意 ☐满意 ☐一般 ☐不满意 改进建议_____

您对本书的总体满意度：

从语言质量角度看 ☐很满意 ☐满意 ☐一般 ☐不满意

从科技含量角度看 ☐很满意 ☐满意 ☐一般 ☐不满意

本书最令您满意的是：

☐指导明确 ☐内容充实 ☐讲解详尽 ☐实例丰富

您认为本书在哪些地方应进行修改？（可附页）

您希望本书在哪些方面进行改进？（可附页）

电子教案支持

敬爱的教师：

为了配合本课程的教学需要，本教材配有配套的电子教案（素材），有需求的教师可以与我们的联系，我们将向使用本教材进行教学的教师免费赠送电子教案（素材），希望有助于教学活动的开展。相关信息请拨打电话 010-62776969 或发送电子邮件至 jsjic@tup.tsinghua.edu.cn 咨询，也可以到清华大学出版社主页（<http://www.tup.com.cn> 或 <http://www.tup.tsinghua.edu.cn>）上查询。



普通高等教育“十一五”国家级规划教材 21世纪大学本科计算机专业系列教材

近期出版书目

- 计算概论(第2版)
- 计算机导论(第2版)
- 程序设计导引及在线实践
- 程序设计基础(第2版)
- 程序设计基础习题解析与实验指导
- C++ 程序设计(第2版)
- 面向对象程序设计(第2版)
- 离散数学(第2版)
- 离散数学习题解答与学习指导(第2版)
- 数据结构与算法
- 数据结构(STL 框架)
- 形式语言与自动机理论(第2版)
- 形式语言与自动机理论教学参考书(第2版)
- 数字逻辑
- 计算机组成原理(第2版)
- 计算机组成原理教师用书(第2版)
- 计算机组成原理学习指导与习题解析(第2版)
- 计算机组成与系统结构
- 计算机组成与体系结构(第2版)
- 微型计算机系统与接口(第2版)
- 计算机系统结构教程
- 计算机系统结构学习指导与题解
- 计算机操作系统
- 计算机操作系统学习指导与习题解答
- 数据库系统原理
- 编译原理
- 软件工程
- 计算机图形学
- 计算机网络(第2版)
- 计算机网络教师用书(第2版)
- 计算机网络实验指导书(第2版)
- 计算机网络习题集与习题解析(第2版)
- 计算机网络软件编程指导书
- 计算机网络工程(第2版)
- 计算机网络工程实验教程
- 人工智能
- 多媒体技术原理及应用(第2版)
- 算法设计与分析(第2版)
- 算法设计与分析习题解答(第2版)
- 信息安全原理及应用

第 1 章 程序设计概论	1
1.1 问题求解	1
1.1.1 思维与语言	1
1.1.2 算法	3
1.1.3 数据结构	5
1.1.4 程序	7
1.1.5 程序设计的一般过程	7
1.2 程序设计语言	8
1.2.1 发展历史	8
1.2.2 程序设计语言的实现	12
1.2.3 高级语言程序基本结构	14
1.2.4 高级语言的语法规则	17
1.2.5 程序的编辑、编译和运行	23
1.3 程序设计范式的演化	25
1.3.1 结构化程序设计	25
1.3.2 函数式程序设计	26
1.3.3 面向对象程序设计	27
1.3.4 事件驱动的程序设计	28
1.3.5 逻辑式程序设计	28
习题 1	30
第 2 章 算法设计基础	34
2.1 算法的描述	34
2.1.1 自然语言方式	34
2.1.2 伪代码方式	35
2.1.3 程序流程图方式	37
2.1.4 PAD 图方式	39
2.2 结构化算法设计初步	40
2.2.1 算法描述	40

2.2.2 算法设计	42
2.3 算法的计算复杂性	47
2.4 常用算法设计策略	48
2.4.1 分治法	48
2.4.2 递归技术	49
2.4.3 贪心法*	50
2.4.4 回溯法*	51
习题2	52
第3章 基本数据类型	54
3.1 数据对象	54
3.1.1 数据的存储	55
3.1.2 生存期与值	55
3.2 基本数据类型	56
3.2.1 整型与浮点型	57
3.2.2 字符类型	58
3.2.3 逻辑型	58
3.2.4 空值型	58
3.3 变量与常量	59
3.3.1 常量	59
3.3.2 变量	61
3.4 指针类型	62
3.4.1 指针定义	62
3.4.2 指针的初始化	63
3.4.3 指针运算	64
3.4.4 指针与常量	65
3.5 引用类型	66
3.6 类型的意义	67
习题3	67
第4章 操作及其控制	70
4.1 基本操作与运算符	70
4.1.1 赋值操作	70
4.1.2 算术操作	70
4.1.3 增1(减1)操作	71
4.1.4 关系、逻辑操作	71
4.1.5 条件运算符	71
4.2 表达式	72
4.2.1 隐含类型转换	72

4.2.2	强制类型转换	73
4.2.3	表达式内的顺序控制	73
4.2.4	运算符的优先级	74
4.2.5	运算符的结合性	74
4.3	语句	75
4.3.1	表达式语句	75
4.3.2	语句间顺序控制	76
4.3.3	复合语句	76
4.3.4	选择语句	76
4.3.5	循环语句	80
4.3.6	直接顺序控制	85
4.4	预处理指令*	87
4.4.1	文件包含指令	88
4.4.2	宏定义指令	88
4.4.3	条件编译指令	91
习题 4		94
第 5 章	函数与数据控制	103
5.1	函数定义与说明	103
5.1.1	函数定义	103
5.1.2	函数说明	104
5.1.3	函数的简单调用	106
5.1.4	函数的重载	106
5.2	数据控制	108
5.2.1	参数传递机制	108
5.2.2	数据对象的引用*	112
5.2.3	作用域	114
5.3	函数的顺序控制	119
5.3.1	函数执行模型*	119
5.3.2	基于栈的实现*	122
5.3.3	函数的自嵌套调用	122
5.4	相关的其他语法	123
5.4.1	数组作函数参数	123
5.4.2	函数指针	126
5.5	常用系统函数	129
5.5.1	数学函数	129
5.5.2	字符串处理函数	129
习题 5		130

第 6 章 构造类型	138
6.1 数组类型	138
6.1.1 数组定义	138
6.1.2 数组的赋值与使用	139
6.1.3 数组的存储	139
6.1.4 数组的初始化	140
6.1.5 字符数组	140
6.1.6 指针与数组	141
6.2 枚举类型	143
6.2.1 枚举类型定义	143
6.2.2 枚举变量定义	143
6.2.3 类型转换	144
6.3 结构类型	145
6.3.1 结构的定义	145
6.3.2 结构的使用	145
6.3.3 结构的实现	147
6.4 结构化数据	148
6.4.1 单链表	148
6.4.2 栈	150
6.4.3 存储的动态分配*	153
6.5 抽象数据类型*	157
6.5.1 结构的扩展	157
6.5.2 C++ 的抽象数据类型	158
6.5.3 C++ 数据类型的层次	159
6.6 自定义类型	160
习题 6	160
第 7 章 结构化算法的实现	167
7.1 基本控制结构的 C++ 实现	167
7.1.1 顺序结构的 C++ 实现	167
7.1.2 分支结构的 C++ 实现	168
7.1.3 循环结构的 C++ 实现	170
7.1.4 复杂结构 C++ 实现示例	171
7.2 子算法设计与 C++ 实现	172
7.2.1 参数为普通类型的子算法	173
7.2.2 参数为指针的子算法	174
7.2.3 参数为引用的子算法	177
7.2.4 子算法设计与 C++ 实现示例	178

7.3 递归与迭代	182
7.3.1 递归	182
7.3.2 迭代	184
7.3.3 应用示例	186
习题 7	189
第 8 章 程序开发过程	192
8.1 软件开发方法概述	193
8.1.1 软件生存周期	193
8.1.2 软件开发方法	193
8.2 软件设计	194
8.2.1 概要设计	195
8.2.2 详细设计	196
8.3 软件编码	196
8.3.1 程序设计方法	196
8.3.2 程序设计语言	197
8.3.3 编码风格	197
8.4 软件测试与调试	198
8.4.1 调试工具及使用	199
8.4.2 调试过程	200
8.4.3 错误类型	200
8.4.4 异常处理*	203
8.4.5 软件测试	207
8.5 程序运行效率	209
8.5.1 适当的算法	209
8.5.2 选择快速运算	210
8.5.3 函数	213
习题 8	213
第 9 章 面向对象程序设计	218
9.1 面向对象方法学*	218
9.1.1 人类认知规律	218
9.1.2 面向对象基本原理	220
9.1.3 认知规律与面向对象	220
9.1.4 面向对象软件开发	221
9.2 基本概念	221
9.2.1 对象	222
9.2.2 类	226
9.2.3 结构与关系	228

9.3 面向对象分析	233
9.3.1 主要原则	233
9.3.2 静态结构分析与对象模型	235
9.3.3 动态行为分析	238
9.4 面向对象设计	241
习题 9	241
第 10 章 类与对象的实现	243
10.1 C++ 中的类	243
10.1.1 类的说明	243
10.1.2 类的定义	245
10.2 对象及其成员的使用	247
10.2.1 对象实例	247
10.2.2 对象成员的引用	248
10.2.3 生命周期	250
10.2.4 类型转换*	255
10.2.5 其他应用	259
10.3 类的特殊成员	268
10.3.1 静态成员*	268
10.3.2 const 成员	271
10.3.3 内联函数	273
10.3.4 友元	273
习题 10	275
第 11 章 类间关系的实现	284
11.1 一般-特殊关系的实现	284
11.1.1 类的继承与派生	284
11.1.2 赋值兼容规则	293
11.1.3 二义性与作用域分辨	294
11.2 多态性与虚函数	300
11.2.1 编译时刻的多态性	300
11.2.2 运行时刻的多态性	301
11.2.3 虚函数	303
11.2.4 纯虚函数与抽象类	310
11.3 整体-部分关系的实现	314
11.4 关联关系的实现	317
11.5 关于类层次的总结*	318
11.5.1 认知规律与类层次	318
11.5.2 构造函数的一般形式	319

11.5.3 成员函数的特征.....	319
习题 11	320
第 12 章 面向对象程序设计的其他技术	325
12.1 运算符重载.....	325
12.1.1 友元运算符.....	326
12.1.2 类运算符.....	328
12.1.3 友元及类运算符的应用.....	329
12.1.4 ++和--运算符的重载.....	330
12.2 流类库.....	333
12.2.1 格式化输入输出.....	334
12.2.2 运算符<<和>>的重载	335
12.2.3 流式文件.....	336
12.2.4 流错误.....	347
12.3 模板.....	348
12.3.1 函数模板.....	348
12.3.2 类模板.....	351
习题 12	354
附录 A Microsoft C++ 的保留字	358
附录 B C++ 中的基本数据类型.....	359
附录 C 扩展 ASCII 字符集	360
附录 D C++ 的运算符	362
参考文献	364

第 1 章

程序设计概论

程序设计以算法设计和数据结构为基础,涉及编码(程序编制)、调试、测试等多方面的理论和技术,本章介绍程序设计的发展历史、程序设计与算法及数据结构之间的关系、高级语言的基本语法特征。

作为全书的导引,本章提前使用了一些概念,读者对此只需做一般性了解,待深入学习后面各章节后会有深刻理解。

1.1 问题求解

计算机本身并不具有人类的智能,利用计算机解决问题时人们必须“告诉”计算机某时刻或某条件下具体做什么。人类与计算机交流的最常用工具就是程序设计语言,人们首先将具体问题抽象为数学模型,然后根据该数学模型,使用程序设计语言描述计算机求解该问题过程中需要做的一系列操作,这种描述就称为**程序**。将程序交给计算机,计算机就会按照程序中规定的具体操作序列对相关数据进行处理,直至得到最终结果从而给出问题的解。

到目前为止,使用计算机解决问题的过程基本都是在模拟人类大脑的解题过程,因此在介绍如何利用计算机解决问题之前,有必要先了解人类大脑是如何解决问题的,重点是在解决问题过程中人脑的思维活动过程。

1.1.1 思维与语言

思维是人脑的机能和产物,是人借助语言、符号、形象来间接地、概括地反映事物本质和规律性的复杂生理与心理活动。形象思维和抽象思维是思维的两种最常见基本方式。所谓**形象思维**是指人依靠大脑中一系列相互关联的具体事物形象(事物在人脑中的映象)的转换来引导思维活动,从而认知和理解事物的过程;而**抽象思维**是指人类通过比较、分析、综合和概括等手段,将隐藏在事物中内在的、本质的、共性的、必然的性质抽取出来,舍弃掉事物表面的、非本质的、偶然的性质,并用概念、范畴和规律等形式固定下来,以揭示、反映客观事物的内在本质和规律性。这里,不准备花费过多的篇幅去阐释两种基本思维方式各自的内涵及关系,只是通过一个实例使读者能够对其有所了解:

桌上已经有 1 个苹果,又放上去 1 个,问一共有几个苹果?

人们可能有下列几种不同的思维过程和方式。

(1) 大脑对眼睛所看见的前后 2 个苹果进行计数后回答 2 个。这就是形象思维。

(2) 当你先写下 1, 然后再写上 +1, 而 $1+1=2$, 因此是 2 个苹果。这就是一种抽象思维(称为形式思维), 是利用已经有加法规则推理后得到结论。

(3) 如果你通过形象思维得到是 2 个苹果; 把苹果换做鸭梨时则是 2 个鸭梨; 想到把鸭梨换做其他的物体也是 2 个, 所以得到 $1+1=2$ 的结论。这也是一种抽象思维(称为辩证思维), 是一种由特殊到一般的抽象。

上述第三种思维方式对程序设计而言是最具意义的, 它给出的是一类问题的具体解决方法——算法。这个过程是一种创造, 从个体经过抽象得到普遍规律, 而通常人们是无法不经思维直接得到这种规律的。为直观地了解人类的思维和解题过程, 下面再来看一个实例:

将正整数数列 9, 6, 7, 3, 4 按照从小到大的顺序排列

显然, 只要是稍有一点数学经验的人就可以完成这个任务。即便是换另外一组正整数, 人们凭借已有的数学知识做这种工作游刃有余。然而, 如果要人们说出是如何解决这个问题, 进一步说出如何解决这一类问题(有限个任意正整数的排序), 则不是一件轻松的事! 为回答这种关于规律性的问题, 按照下面的步骤, 换一个角度来分析上面的问题。

首先给出上述排序问题的抽象形式: 将任意一组正整数 $a_1, a_2, a_3, \dots, a_n$ 按照从小到大的顺序排列。这是一个较上面的排序问题复杂得多的另一个问题, 它给出了正整数排序问题的抽象描述, 它的解决将给出一种正整数排序的规律, 任何一组具体正整数的排序都是它的一个特例, 因此都可以使用这种规律加以解决。下面以这个排序问题为例, 根据人脑的思维过程, 探讨如何从一个具体问题的解题过程得到一般性问题的解。

由于对 9, 6, 7, 3, 4 而言, 这些“符号”代表了具体的数值, 这种具有不同值的特性使人们在排序时可以很容易地通过形象思维, 根据各个不同数值之间的关系, 利用简单的逻辑思维解决一个具体的排序问题。但是对于 $a_1, a_2, a_3, \dots, a_n$ 而言, 虽然它们代表的也是数字, 但这些数字是不确定的, 使用这些代表不确定数字的符号解决排序问题将给出排序问题的通用解。由于符号所代表数据的不确定性使得各符号之间的关系很难把握, 因此给这类问题的解决带来了难度, 解决这种问题需要有较强的逻辑思维能力, 需要研究者充分发挥自己的想象能力, 借鉴解决具体排序问题时的形象思维过程, 归纳、总结并构造出解决问题的算法。在具体的排序过程中, 有人可能按如下步骤进行。

- ① 交换 9 和 3, 结果形成: 3, 6, 7, 9, 4。
- ② 交换 6 和 4, 结果形成: 3, 4, 7, 9, 6。
- ③ 交换 7 和 6, 结果形成: 3, 4, 6, 9, 7。
- ④ 交换 9 和 7, 完成排序: 3, 4, 6, 7, 9。

从形式上看不出这种排序过程具有普遍性, 但是仔细分析上述过程的细节可以从中抽象出具有普遍性的排序方法: 为什么第一步要交换 9 和 3 呢? 为什么第二步交换 6 和 4 呢? ..., 为什么完成第四步后就结束了呢?

显然用 9 和 3 交换是因为 3 是最初数据中最小的数, 应该排在所有数据的前面; 类似地, 6 和 4 的交换是由于在第一步形成的结果中, 4 是数字 3 后面 4 个未排序数字中的最小数, 应该排在这些未排序数字前面, 以此类推。当最后只剩一个未排序数字时排序过程结束。这种讨论虽然使用了具体的数字, 然而交换数字发生在第一个未排序数字与未排序数