



普通高等教育“十一五”国家级规划教材

21世纪高等学校计算机专业核心课程规划教材

算法设计与分析

(第3版)

吕国英 李茹 王文剑 任瑞征 钱宇华 编著



清华大学出版社

普通高等教育“十一五”国家级规划教材
21 世纪高等学校计算机专业核心课程规划教材

算法设计与分析(第 3 版)

吕国英 李 茹 王文剑 任瑞征 钱宇华 编著

清华大学出版社
北京

内 容 简 介

本教材的内容遵循《高等学校计算机科学与技术专业公共核心知识体系与课程》(教育部高等学校计算机科学与技术教学指导委员会,2008)的知识体系,介绍算法及其设计、分析的基础知识,并通过大量例题,讲解枚举法、递推法、分治法、贪婪算法、动态规划及与图搜索有关的算法策略。除此之外,还对算法设计基本工具的使用和算法设计中的技巧做了讲解。最后通过案例的一题多解进行算法设计的实践。算法采用了接近自然语言(英语)的符号,可读性强,适合于不同程序设计语言背景的读者学习。

本书可以作为高等院校计算机及其相关专业高年级本科生和研究生算法设计课程的教材,也可作为计算机工作者、广大程序设计爱好者和信息学爱好者的参考书。

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

算法设计与分析/吕国英等编著. —3版. —北京:清华大学出版社,2015

21世纪高等学校计算机专业核心课程规划教材

ISBN 978-7-302-39175-3

I. ①算… II. ①吕… III. ①电子计算机—算法设计—高等学校—教材 ②电子计算机—算法分析—高等学校—教材 IV. ①TP301.6

中国版本图书馆CIP数据核字(2015)第017677号

责任编辑:闫红梅

封面设计:杨 兮

责任校对:梁 毅

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

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

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.5 字 数:489千字

版 次:2006年3月第1版 2015年6月第3版 印 次:2015年6月第1次印刷

印 数:39501~41500

定 价:39.00元

产品编号:061983-01

进入 21 世纪,各国高科技发展突飞猛进,对教育资源、人才资源的争夺也日益激烈,计算机软件开发人才更是处于核心竞争地位。培养应用型软件开发人才成为提高国家科技实力的重要步骤。国家 973 信息技术与高性能软件基础规划项目首席科学家顾钧教授和中国工程院院士李国杰教授指出:“我国的软件开发要算法先行,这样才能推动软件技术的研究与开发,提高我国企业软件产品的技术竞争力和市场竞争力。”

“算法设计与分析”是一门理论性与实践性相结合的课程,是计算机科学与计算机应用专业的核心课程。学习算法设计可以在分析解决问题的过程中,培养学生抽象思维和缜密概括的能力,提高学生的软件开发设计能力。

全书共包含四篇:

1. 第 1 篇“引入篇”共两章,从认识算法开始,介绍问题求解的步骤及算法在其中的重要地位,讲解了算法效率分析的基本方法,对当前常用的算法软件进行了简要概述(此节可作为选修)。

2. 第 2 篇“基础篇”,对算法的重复操作机制——循环和递归的设计要点、算法中数据结构的选择和提高算法效率的基本技巧做了讲解,这些都是算法设计的重要基础。

3. 第 3 篇“核心篇”共两章,主要介绍了几种常用的算法策略,如:枚举法、递推法、分治法、贪婪算法、动态规划及与图搜索有关的算法策略,并对算法策略进行了总结比较。

4. 第 4 篇“应用篇”共两章,第 6 章通过随机序列改进前面介绍的算法效率,介绍概率经典算法(第 3 版新增内容)。第 7 章以问题为节,每节中针对同一问题给出采用不同的数学模型、不同数据结构或不同的算法策略进行算法设计,并进行效率分析。这部分内容是对算法设计学习的实践。

本教材建设的理念是“实用、适用”。书中的例题选择力求简单但具有代表性,从分析问题开始,经模型建立,再进行算法设计(包括数据结构设计)和算法分析。这样做有利于培养学生“设计”算法的能力,而不是“记忆”算法的能力。并力争浅显易懂地讲解较深奥的算法设计策略和算法分析方法。

本书的主要特点有:

1) 重系统性

教材的第 3 篇“核心篇”摒弃同类教材中根据问题划分章节的方法,通过对算法策略特点的概括和归纳,以同一策略下的应用差别来划分章节,使得教材结构更合理、讲解更系统,更加符合认知规律。同时,在各章末尾对算法进行比较、总结,使学生能方便、全面地掌握算法策略的本质及算法应用体系。

2) 重启发性

本书中例题都是经过问题分析、数学建模、数据结构设计后,才给出算法设计和算法分析的。这样讲解富有启发性,不仅培养了学生算法设计的思维方式,还能改变学生被动接受

知识的习惯,养成主动学习的意识,进而提高创新能力。

3) 重适用性

第2篇“基础篇”是从程序设计到算法设计承上启下的内容,对问题求解的基本方法、算法基本工具的使用及提高算法效率的基本技巧做了必要的总结、归纳。相信这些内容会对普通院校的广大学生有较大的裨益,促进其打好学习算法设计的基础。弥补了以往教材缺乏课程间衔接内容的缺陷,可以增强学生学习该课程的自信心,提高教学效率。

4) 重开放性

教材在第1篇中对现代算法进行了概览,旨在扩大学生的知识面,提高其对算法设计的学习兴趣。教材还独特地介绍了从算法到程序转换的要点,引导学生不是仅仅停留在形式化的算法描述阶段,而是大胆上机实现,提高学生学习本学科的兴趣。

5) 重实践性

第4篇“应用篇”是本教材的一大亮点。该篇以问题为节,每节针对同一问题采用不同的数学模型、不同的数据结构或不同的算法策略进行算法设计,扩展学生解决问题的思路,学会灵活运用算法知识,而不是生搬硬套教材中的算法。同时,也可以通过对多种算法设计的分析比较认识算法的优劣,从而设计出质量优良的算法。

在学习算法设计的过程中,有的读者一定会感到所学的内容和大多例题都离现实问题很远,似乎用途不大。这是因为现实中的实际问题往往比较复杂,需要具备丰富的领域知识、算法设计方法和技巧规范及软件工程的开发规范等综合技能。所以,只能通过一些简单、抽象的例子,对基础的算法策略进行讲解。待打好算法设计基础且有足够的问题领域知识储备后,才能去解决实际应用问题。附录“算法设计与分析课程设计大纲”中给出一些与现实结合相对较紧的练习,区别于章节习题,希望读者广开思路,应用所学知识解决问题。

《算法设计与分析》自2006年第1版出版以来,受到读者的广泛好评,多所院校将本书作为“算法设计与分析”课程的教材,在此,我们对读者表示由衷的感谢!同时,我们深感重任在身,在听取广大读者提出的宝贵意见的基础上,极为慎重地对待本次改版工作。教材的出版凝聚了出版社工作人员的辛勤汗水,在此感谢出版社领导与编辑们的信任与付出。

随着信息化时代的到来,计算机开发平台日新月异,计算机的应用也不断拓展到了各个领域;各类算法和技巧层出不穷,本书只能是管中窥豹。若能达到本书的初衷——使读者能掌握算法设计的基本方法和技巧,打好软件开发的基础,我们就深感满意了。

由于作者水平有限,书中不当之处敬请专家和读者指正。

作 者

2014年10月

第1篇 引入篇

| | |
|-------------------------|----|
| 第1章 算法概述 | 3 |
| 1.1 用计算机求解问题与算法 | 3 |
| 1.1.1 用计算机求解问题的步骤 | 3 |
| 1.1.2 算法及其要素和特性 | 5 |
| 1.1.3 算法设计及基本方法 | 7 |
| 1.1.4 从算法到实现 | 10 |
| 1.2 算法设计步骤及描述 | 12 |
| 1.2.1 算法描述简介 | 12 |
| 1.2.2 本书算法描述约定 | 16 |
| 1.2.3 一个简单问题的求解过程 | 19 |
| 1.3 现代常用算法概览 | 21 |
| 1.3.1 压缩算法 | 21 |
| 1.3.2 加密算法 | 24 |
| 1.3.3 人工智能算法 | 26 |
| 1.3.4 并行算法 | 28 |
| 1.3.5 其他实用算法 | 30 |
| 第2章 算法分析基础 | 35 |
| 2.1 算法分析体系及计量 | 35 |
| 2.1.1 算法分析的评价体系 | 35 |
| 2.1.2 算法的时间复杂性 | 36 |
| 2.1.3 算法的空间复杂性 | 39 |
| 2.1.4 NP 完全问题 | 40 |
| 2.2 算法分析实例 | 41 |
| 2.2.1 非递归算法分析 | 41 |
| 2.2.2 递归算法分析 | 43 |
| 2.2.3 提高算法质量 | 46 |

第2篇 基础篇

| | |
|------------------------------|-----|
| 第3章 算法基本工具和优化技巧 | 51 |
| 3.1 循环与递归 | 51 |
| 3.1.1 循环设计要点 | 52 |
| 3.1.2 递归设计要点 | 57 |
| 3.1.3 递归与循环的比较 | 61 |
| 3.2 算法与数据结构 | 67 |
| 3.2.1 原始信息与处理结果的对应存储 | 69 |
| 3.2.2 数组使信息有序化 | 72 |
| 3.2.3 数组记录状态信息 | 74 |
| 3.2.4 大整数存储及运算 | 77 |
| 3.2.5 构造趣味矩阵 | 80 |
| 3.2.6 一维与二维的选择 | 86 |
| 3.3 优化算法的基本技巧 | 89 |
| 3.3.1 算术运算的妙用 | 89 |
| 3.3.2 标志量的妙用 | 92 |
| 3.3.3 信息数字化 | 97 |
| 3.4 优化算法的数学模型 | 104 |
| 3.4.1 杨辉三角形的应用 | 106 |
| 3.4.2 最大公约数的应用 | 107 |
| 3.4.3 公倍数的应用 | 110 |
| 3.4.4 斐波那契数列的应用 | 111 |
| 3.4.5 特征根求解递推方程 | 112 |
| 习题 | 114 |

第3篇 核心篇

| | |
|--------------------------|-----|
| 第4章 基本的算法策略 | 121 |
| 4.1 迭代算法 | 121 |
| 4.1.1 递推法 | 122 |
| 4.1.2 倒推法 | 125 |
| 4.1.3 迭代法解方程 | 128 |
| 4.2 蛮力法 | 131 |
| 4.2.1 枚举法 | 131 |
| 4.2.2 其他范例 | 133 |
| 4.3 分而治之算法 | 136 |

| | | |
|------------|------------------|------------|
| 4.3.1 | 分治算法框架 | 137 |
| 4.3.2 | 典型二分法 | 138 |
| 4.3.3 | 二分法不相似情况 | 140 |
| 4.3.4 | 二分法不独立情况 | 143 |
| 4.3.5 | 非等分分治 | 147 |
| 4.4 | 贪婪算法 | 151 |
| 4.4.1 | 可绝对贪婪问题 | 151 |
| 4.4.2 | 相对或近似贪婪问题 | 159 |
| 4.4.3 | 贪婪策略算法设计框架 | 161 |
| 4.5 | 动态规划 | 162 |
| 4.5.1 | 认识动态规划 | 162 |
| 4.5.2 | 动态规划算法设计框架 | 166 |
| 4.5.3 | 突出阶段性的动态规划应用 | 167 |
| 4.5.4 | 突出递推的动态规划应用 | 176 |
| 4.6 | 算法策略间的比较 | 182 |
| 4.6.1 | 不同算法策略特点小结 | 182 |
| 4.6.2 | 算法策略间的关联 | 183 |
| 4.6.3 | 算法策略侧重的问题类型 | 185 |
| | 习题 | 185 |
| 第5章 | 图的搜索算法 | 188 |
| 5.1 | 图搜索概述 | 188 |
| 5.1.1 | 图及其术语 | 188 |
| 5.1.2 | 图搜索及其术语 | 192 |
| 5.2 | 广度优先搜索 | 192 |
| 5.2.1 | 算法框架 | 193 |
| 5.2.2 | 广度优先搜索的应用 | 194 |
| 5.3 | 深度优先搜索 | 198 |
| 5.3.1 | 算法框架 | 198 |
| 5.3.2 | 深度优先搜索的应用 | 199 |
| 5.4 | 回溯法 | 207 |
| 5.4.1 | 认识回溯法 | 207 |
| 5.4.2 | 算法简介算法框架 | 211 |
| 5.4.3 | 应用1——基本的回溯搜索 | 213 |
| 5.4.4 | 应用2——排列及排列树的回溯搜索 | 217 |
| 5.4.5 | 应用3——最优化问题的回溯搜索 | 221 |
| 5.5 | 分支限界法 | 224 |
| 5.5.1 | 分支搜索算法 | 224 |
| 5.5.2 | 分支-限界搜索算法 | 231 |

| | |
|--------------------|-----|
| 5.5.3 算法框架 | 237 |
| 5.6 图的搜索算法小结 | 238 |
| 习题 | 239 |

第4篇 应用篇

| | |
|-----------------------------------|-----|
| 第6章 概率算法 | 245 |
| 6.1 概述 | 245 |
| 6.2 统计模拟——蒙特卡罗算法 | 246 |
| 6.2.1 数值计算方法——蒙特卡罗算法 | 247 |
| 6.2.2 考虑正确几率的算法——蒙特卡罗算法 | 249 |
| 6.3 随机序列提高算法的平均复杂度——舍伍德算法 | 251 |
| 6.4 随机生成答案并检测答案正确性——拉斯维加斯算法 | 253 |
| 第7章 算法设计实践 | 256 |
| 7.1 循环赛日程表(4种) | 256 |
| 7.2 求3个数的最小公倍数(4种) | 262 |
| 7.3 猴子选大王(4种) | 266 |
| 7.4 最大子段和问题(5种) | 270 |
| 7.5 背包问题(11种) | 275 |
| 7.5.1 与利润无关的背包问题 | 275 |
| 7.5.2 与利润有关的背包问题 | 279 |
| 7.6 主元素问题(6种) | 293 |
| 附录 算法设计与分析课程设计大纲 | 297 |

第 1 篇

引 入 篇

本篇内容：

第1章 算法概述

第2章 算法分析基础

1.1 用计算机求解问题与算法

问题求解(problem solving)是个大课题,它涉及归约、推断、决策、规划、常识推理、定理证明、相关过程等核心概念。人工智能是这个课题下的一个分支,人工智能的第一个大成就是进一步开发了能够求解难题的下棋(如国际象棋)程序。把在下棋程序中应用的某些技术,发展成为搜索和问题归约这样的人工智能基本算法。今天的计算机程序能够下锦标赛水平的各种方盘棋,如五子棋和国际象棋。有些软件甚至还能够用自动总结的经验来改善软件自身性能。由此可以理解“问题求解”的重点是要制造智能计算机,以便模拟人的智能去进行问题求解,属于尖端科技。而一般计算机面对现实问题是无能为力的,需要人类对问题抽象化、形式化后才能机械地执行,学习算法设计的重点就是把人类找到的求解问题的方法、步骤以过程化、形式化、机械化的形式表示出来,以便让计算机执行(当然人工智能软件系统也离不开“算法设计”这个最基本的软件设计环节)。本书就把学习的目标定为“用计算机求解问题”。

1.1.1 用计算机求解问题的步骤

人类在解决一个问题时,根据不同的经验、不同的环境会采用不同的方法,用计算机解决现实中的问题,同样也有很多不同的方法,但解决问题的基本步骤是相同的。下面给出用计算机求解问题的一般步骤。

1. 问题分析

准确、完整地理解和描述问题是解决问题的第一步。要做到这一点,必须注意以下一些问题:在未经加工的原始表达中,所用的术语是否都明白其准确定义?题目提供了哪些信息?这些信息有什么用?题目要求得到什么结果?题目中有哪些假定?是否有潜在的信息?判定求解结果所需要的中间结果有哪些?等等。针对每个具体的问题,必须认真审查问题描述,理解问题的真实要求。

2. 数学模型建立

用计算机解决实际问题必须有合适的数学模型。对一个实际问题建立数学模型,可以考虑这样两个基本问题:最适合于此问题的数学模型是什么?是否有已经解决了的类似问

题可借鉴?

如果上述第二个问题的答复是肯定的,那么通过类似问题的分析、比较和联想,可加速问题的解决。但上述第一个问题毕竟是更重要的。如何选择恰当的数学工具来表达已知的和要求的量,受多种因素影响:设计人员的数学知识水平,已知的数学模型是否表达方便,计算是否简单,所要进行的操作种类的多少与功能的强弱等。同一问题可以用不同的数学工具建立不同的模型,因此要对不同的模型进行分析、比较,从中选出最有效的模型。然后根据选定的数学模型,对问题进行重新描述。

此时,应考虑下列一些问题:模型能否清楚地表示与问题有关的所有重要信息?模型中是否存在与所期望的结果相关的数学量?能否正确反映输入、输出的关系?用计算机实现该模型是否有困难?如能取得满意的回答,那么该数学模型可作为候选模型。

3. 算法设计与选择

算法设计是指设计求解某一特定类型问题的一系列步骤,而这些步骤是可以通过计算机的基本操作来实现的。算法设计要同时结合数据结构的设计,简单地讲数据结构的设计就是选取存储方式,如确定问题中的信息是用数组存储还是用普通变量存储(或“数据结构”课程中介绍的更多存储方式)。因为,不同的数据结构的设计将导致差异很大的算法。算法的设计与模型的选择更是密切相关的,但同一模型仍然可以有不同的算法,而且它们的有效性可能有相当大的差距。选择方法和模型建立大致相同,首先考虑学过的方法是否可以借鉴?最适合于此问题的算法是什么?

4. 算法表示

对于复杂的问题,确定算法后可以通过图形准确表示算法。算法的表示方式很多,如算法流程图、盒图、PAD图和伪码(类似于程序设计语言)。本书对简单的算法将不进行图形表示。

5. 算法分析

算法分析的目的,首先是为了对算法的某些特定输入,估算该算法所需的内存空间和运行时间;其次是为了建立衡量算法优劣的标准,用以比较同一类问题的不同算法。通常将时间和空间的增长率作为衡量的标准,参见 1.1.4 节算法及其设计的评价。

6. 算法实现

求解某一特定类型问题的算法设计完成,并证明其正确性之后,就要根据算法编制计算机程序来实现它。在编制程序之前,还要选取存储类型,用来表达所用模型的各个方面。因此,根据选用的程序设计语言,要解决下列一些问题:有哪些变量,它们是什么类型?需要多少数组、规模有多大?用什么结构来组织数据?需要哪些子程序?等等。

算法的实现方式,对运算速度和所需内存容量都有很大影响。

7. 程序测试及调试

目前,程序的正确性尚未得到根本的解决,软件测试仍是发现软件错误和缺陷的主要手

段。软件测试是将编制的程序投入实际运行前,用手工或编译程序等方法进行测试,是发现语法错误和逻辑错误的过程。调试就是根据测试时所发现的错误,进一步诊断,找出原因和具体的位置进行修正的过程。

8. 结果整理文档编制

编制文档的目的是让人理解编写的程序代码。要把代码编写清楚。代码本身就是文档。同时还要采用注释的方式,另外还包括算法的流程图,自顶向下各研制阶段的有关记录,算法的正确性证明(或论述),程序测试过程、结果,对输入/输出的要求及格式的详细描述。

【提示】 在这些步骤中,哪一步是解决问题的核心?为什么?

本书在例题讲解时,侧重第1~5步。

1.1.2 算法及其要素和特性

1. 算法的定义

算法(algorithm)是指在解决问题时,按照某种机械步骤一定可以得到问题结果(有解时给出问题的解,无解时给出无解的结论)的处理过程。当面临某个问题时,需要找到用计算机解决这个问题的方法和步骤,算法就是解决这个问题的方法和步骤的描述。

所谓机械步骤是指,算法中有待执行的运算和操作,必须是相当基本的。换言之,它们都是能够精确地被计算机运行的算法,执行者(计算机)甚至不需要掌握算法的含义,即可根据该算法的每一步骤要求,进行操作并最终得出正确的结果。

“算法”这个词其实并不是一个陌生的词,因为从小学大家就开始接触算法了。例如做四则运算,必须按照一定的算法步骤一步一步地做。“先运算括号内再运算括号外,先乘除后加减”可以说是四则运算的算法。以后学习的指数运算、矩阵运算和其他代数运算的运算规则都是一种算法。

就本课程而言,算法就是计算机解决问题的过程。在这个过程中,无论是形成解题思路还是编写算法,都是在实施某种算法。前者是推理实现的算法,后者是操作实现的算法。

2. 算法的3要素

算法由操作、控制结构、数据结构3要素组成。

1) 操作

算法实现平台尽管有许多种类,它们的函数库、类库也有较大差异,但必须具备的最基本的操作功能是相同的。这些操作包括以下几个方面。

算术运算:加、减、乘、除。

关系比较:大于、小于、等于、不等于。

逻辑运算:与、或、非。

数据传送:输入、输出(计算)、赋值(计算)。

2) 算法的控制结构

一个算法功能的实现不仅取决于所选用的操作,还取决于各操作之间的执行顺序,即控制结构。算法的控制结构给出了算法的框架,决定了各操作的执行次序。这些结构包括以下几个方面。

顺序结构:各操作是依次执行的。

选择结构:由条件是否成立来决定选择执行。

循环结构:有些操作要重复执行,直到满足某个条件时才结束,这种控制结构也称为重复或迭代结构。

本书认为模块间的调用也是一种控制结构,特别地模块自身的直接或间接调用是递归结构,是一种功能很强的控制重复的结构。在3.1节将重点介绍利用循环、递归机制设计算法中重复操作的要点。

3) 数据结构

算法操作的对象是数据,数据间的逻辑关系、数据的存储方式及处理方式就是数据的数据结构。它与算法设计是紧密相关的,在3.2节、第7章中将通过例题进行介绍。

【提示】 算法的要素很有限,这是否能帮助你回答上一个提示当中的问题?并且使你认识到抵制盗版软件,支持正版软件的必要性。

3. 算法的基本性质

进一步理解,算法就是把人类找到的求解问题的方法,经过过程化、形式化后,用以上要素表示出来。在算法的表示中要满足以下的性质:

- 目的性——算法有明确的目的,算法能完成赋予它的功能。
- 分步性——算法为完成其复杂的功能,由一系列计算机可执行的步骤组成。
- 有序性——算法的步骤是有序的,不可随意改变算法步骤的执行顺序。
- 有限性——算法是有限的指令序列,算法所包含的步骤是有限的。
- 操作性——有意义的算法总是对某些对象进行操作,使其改变状态,完成其功能。

4. 算法的地位

算法是计算机学科中最具有方法论性质的核心概念,也被誉为计算机学科的灵魂。

数学大师吴文俊指出:“我国传统数学在从问题出发以解决问题为主旨的发展过程中,建立了以构造性与机械化为其特色的算法体系,这与西方数学以欧几里得《几何原本》为代表的所谓公理化演绎体系正好遥遥相对。……肇始于我国的这种机械化体系,在经过明代以来几百年的相对消沉后,由于计算机的出现,已越来越为数学家所认识与重视,势将重新登上历史舞台。”吴文俊创立的几何定理的机器证明方法(世称吴方法),用现代的算法理论,焕发了中国古代数学的算法传统,享有很高的国际声誉,也受到国家的高度关注。他因此于2001年获得了第一届国家最高科学技术奖。

5. 算法的基本特征

并不是所有问题都有可以解决它们的方法,也不是所有人类解决问题的方法都能设计出相应的算法。算法必须满足以下5个重要特性。

1) 有穷性

一个算法在执行有穷步之后必须结束,也就是说一个算法它所包含的计算步骤是有限的,即算法中的每个步骤都能在有限时间内完成。

2) 确定性

对于每种情况下所应执行的操作,在算法中都有确切的规定,使算法的执行人或阅读者都能明确其含义及如何执行。并且在任何条件下,算法都只有一条执行路径。

3) 可行性

算法中描述的操作都可以通过已经实现的基本操作运算有限次实现之。

4) 算法有零个或多个的输入

有输入作为算法加工对象的数据,通常体现为算法中的一组变量。有些输入量需要在算法执行过程中输入,而有的算法表面上可以没有输入,实际上已被嵌入算法之中。

5) 算法有一个或多个的输出

它是一组与输入有确定关系的量值,是算法进行信息加工后得到的结果。

1.1.3 算法设计及基本方法

算法设计(designing algorithm)作为用计算机解决问题的一个步骤,其任务是对各类具体问题设计出良好的算法。算法设计作为一门课程,是研究设计算法的规律和方法。

在设计算法时,应当严格考虑算法的以下质量指标。

1) 正确性(correctness)

首先,算法对于一切合法的输入数据都能得出满足要求的结果;其次对于精心选择的、典型的、苛刻的几组输入数据,算法也能够得出满足要求的结果。

2) 可读性(readability)

算法主要是为了人的阅读与交流,其次才是让计算机执行。因此算法应该易于人的理解;另一方面,晦涩难读的算法易于隐藏较多错误而难以调试;有些算法设计者总是把自己设计的算法写的只有自己才能看懂,这样的算法反而没有太大的实用价值。

3) 稳健性(robustness)

当输入的数据非法时,算法应当恰当地做出反映或进行相应处理,而不是产生莫名其妙的输出结果。这就需要一定要充分地考虑到可能的异常情况(unexpected exceptions),并且处理出错的方法不应该是简单地中断算法的执行,而应是返回一个表示错误或错误性质的值,以便在更高的抽象层次上进行处理。

4) 高效率与低存储量需求

通常,效率指的是算法执行时间;存储量指的是算法执行过程中所需的最大存储空间。两者都与问题的规模有关。这一点在第2章中详细介绍。

针对不同的问题算法设计方法、策略很多,学习和掌握它们是本书的主要任务,这里先介绍几个算法设计的基本模型。

【提示】 若需要你组织一台新年联欢会,你会怎么做?先确定节目,以节目选人;还是把指定几个人自己选节目,然后把它们协调成一台联欢会?可能还有更好的办法吧?

1. 结构化方法

算法的质量首先取决于它的结构。算法设计和建筑设计极为相似,一座建筑物的整体质量首先取决于它的钢筋混凝土结构是否牢固,然后才是它的外装修质量。同样,一个算法的质量优劣,首先取决于它的结构,其次才是它的速度、界面等其他特性;如果一个程序中的所有模块都只使用顺序、选择和循环3种基本结构,那么不管这个程序中包含多少个模块,它仍然具有清晰的结构。

结构化方法总的指导思想是自顶向下、逐步求精。它的基本原则是功能的分解与模块化。

所谓“自顶向下”是将现实世界的问题经抽象转化为逻辑空间或求解空间的问题;是将复杂且规模较大的问题划分为较小问题,找出问题的关键和重点,然后抽象地、概括地描述问题。

所谓“逐步求精”是将复杂问题经抽象化处理变为相对比较简单的问题。经若干步精化处理,最后细化到用“3种基本结构”及基本操作去描述算法。

所谓“模块化”是指把一个大的程序按照一定的原则划分为若干个相对独立但又相关的小模块(函数)的方法。

结构算法设计技术的优越性:

(1) 自顶向下逐步求精的方法符合人类解决复杂问题的普遍规律,因此可以显著提高软件开发工程的成功率和生产率。

(2) 用先全局后局部、先整体后细节、先抽象后具体的逐步求精过程开发出的算法有清晰的层次结构,因此容易阅读和理解。

2. 面向对象方法

所谓对象(object)是包含数据和对数据操作代码的实体,或者说是在传统的数据结构中加入一些被称为成员函数的过程,因而赋予对象以动作。在面向对象算法设计(object-oriented design)中,对象具有与现实世界的某种对应关系,人们正是利用这种关系对问题进行分解,如图 1-1 所示。

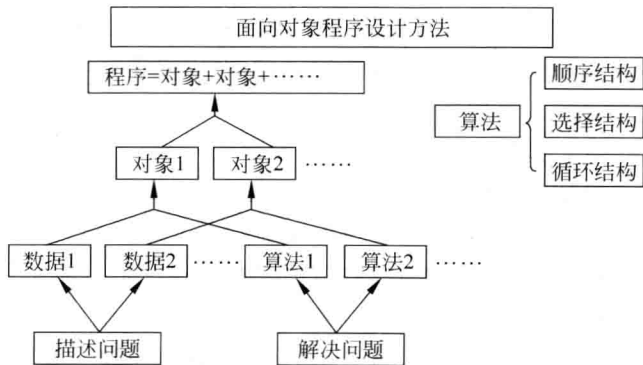


图 1-1 面向对象程序设计方法