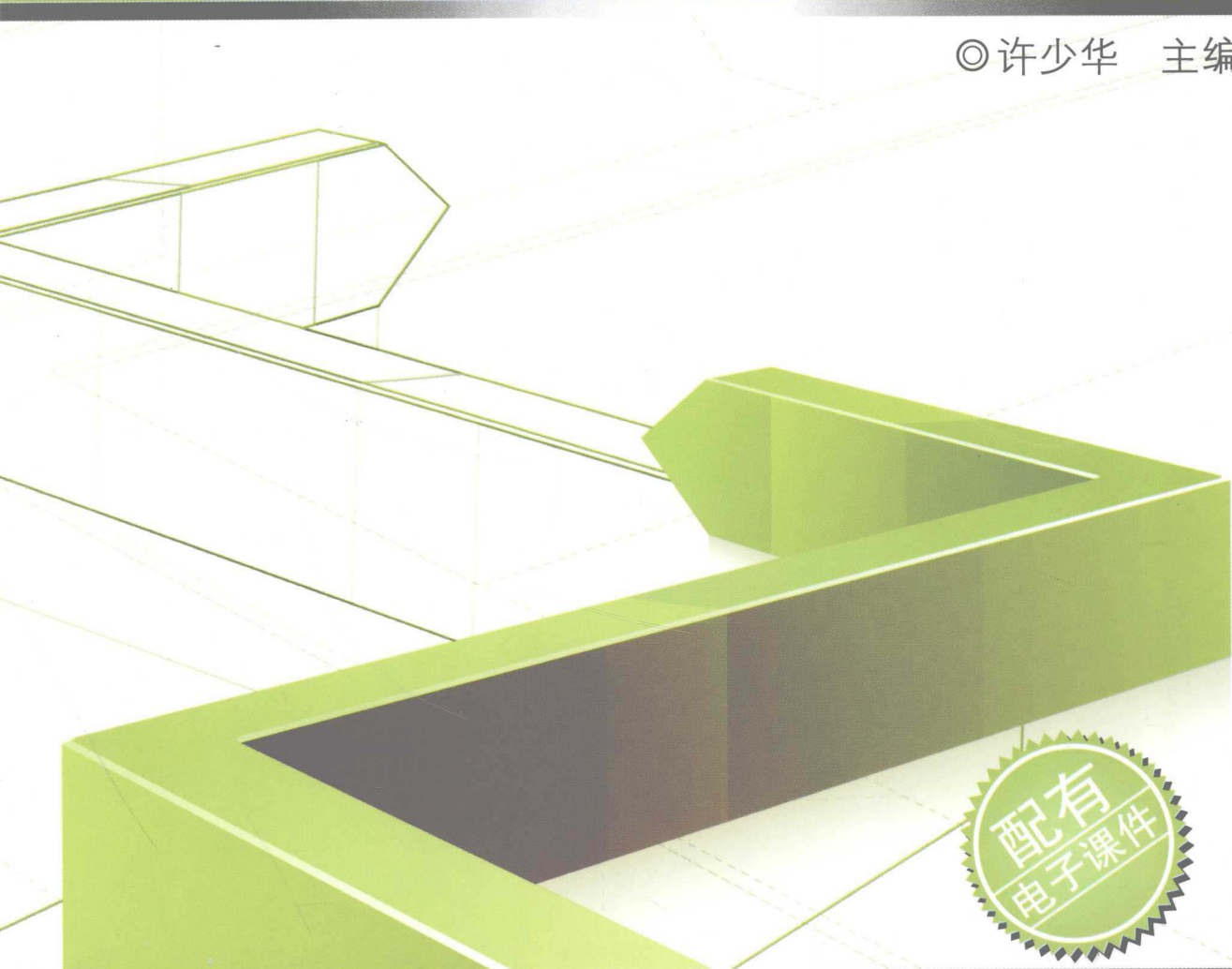


高等学校“十二五”规划教材·计算机软件工程系列

Design and Analysis of Computer Algorithms

算法设计与分析

◎许少华 主编



淡化理论讲解/突出算法设计策略思想/由易到难引出实例
注重同一问题不同算法的比较/详细分析每种算法的优缺点



哈尔滨工业大学出版社
HARBIN INSTITUTE OF TECHNOLOGY PRESS

高等学校“十二五”规划教材·计算机软件工程系列

算法设计与分析

许少华 主 编
富 宇 潘海为 邱兆文 副主编

哈尔滨工业大学出版社

内容简介

本书为大学计算机相关专业核心课程——“算法设计与分析”教材。全书以算法设计策略为知识单元,系统介绍算法设计方法与分析技巧,主要内容包括:算法概述、分治与递归、贪心算法、动态规划、搜索算法、网络流和匹配、线性规划。在介绍每一种方法时,阐述了它的应用背景,并注意与其他方法的比较。

本书结构简明、内容丰富,为突出教材的可读性和可用性,章内设有典型例题分析,章末配有难易适度的习题,有利于读者对相关内容的理解。本书适合于作为大学计算机科学与技术专业、软件工程专业及相关专业本科生和研究生教材,也适合广大工程技术人员学习参考。

图书在版编目(CIP)数据

算法设计与分析/许少华主编. —哈尔滨:哈尔滨工业大学出版社,2011.8
高等学校“十二五”规划教材·计算机软件工程系列
ISBN 978-7-5603-3365-6

I. ①算… II. ①许… 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(2011)第 161578 号

策划编辑 王桂芝 赵文斌

责任编辑 唐 蕾

出版发行 哈尔滨工业大学出版社

社 址 哈尔滨市南岗区复华四道街 10 号 邮编 150006

传 真 0451-86414749

网 址 <http://hitpress.hit.edu.cn>

印 刷 哈尔滨市石桥印务有限公司

开 本 787mm×1092mm 1/16 印张 12.25 字数 300 千字

版 次 2011 年 8 月第 1 版 2011 年 8 月第 1 次印刷

书 号 ISBN 978-7-5603-3365-6

定 价 28.00 元

(如因印装质量问题影响阅读,我社负责调换)

高等学校“十二五”规划教材

计 算 机 软 件 工 程 系 列

编 审 委 员 会

名誉主任 丁哲学

主 任 王义和

副 主 任 王建华

编 委 (按姓氏笔画排序)

王霓虹 印桂生 许少华 任向民

衣治安 刘胜辉 苏中滨 张 伟

苏建民 李金宝 苏晓东 张淑丽

沈维政 金 英 胡 文 姜守旭

贾宗福 黄虎杰 董宇欣

◎ 序

Foreword

随着计算机工程的发展和社会对计算机工程人才需求的增长,软件工程专业培养目标更加明确,特色更加突出。目前,国内多数高校软件工程专业培养目标是以需求为导向,注重培养学生掌握软件工程基本理论、专业知识和基本技能,具备运用先进的工程化方法、技术和工具从事软件系统分析、设计、开发、维护和管理等工作能力,以及具备参与工程项目的实践能力、团队协作能力、技术创新能力和市场开拓能力,具有发展成软件行业高层次工程技术和企业管理人才潜力的,适应社会市场经济和信息产业发展需要的“工程实用型”人才。

本系列教材针对软件工程专业“突出学生的软件开发能力和软件工程素质,培养从事软件项目开发和管理的先进工程技术人才”的培养目标,集9家软件学院(软件工程专业)的优秀作者和强势课程,本着“立足基础,注重实践应用;科学统筹,突出创新特色”的原则,精心策划编写。具体特色如下:

1. 紧密结合企业需求,多校优秀作者联合编写

本系列教材编写在充分进行企业需求、学生需要、教师授课方便等多方市场调研的基础上,采取了校企适度联合编写的做法,根据目前企业的普遍需要,结合在校学生的实际学习情况,校企作者共同研讨、确定课程的安排和相关教材内容,力求使学生在校学习过程中就能熟悉和掌握科学研究及工程实践中需要的理论知识和实践技能,以便适应就业及创业的需要,满足国家软件工程专业人才培养需要。

2. 多门课程系统规划,注重培养学生工程素质

本系列教材精心策划,从计算机基础课程→软件工程基础与主干课程→设计与实践课程,系统规划,统一编写。既考虑到每门课程的相对独立性,基础知识的完整性,又兼顾相关课程之间的横向联系,避免知识点的简单重复,力求形成科学、完整的知识体系。

本系列教材中的《离散数学》、《数据库系统原理》、《算法设计与分析》等基础教材在引入概念和理论时,尽量使其贴近社会现实及软件工程等学科的技术和应用,力图将基本知识与软件工程学科的实际问题结合起来,在具备直观性的同时强调启发性,必须让学生理解所

学的知识。《软件工程导论》、《软件体系结构》、《软件质量保证与测试》、《软件项目管理》等软件工程主干课程以《软件工程导论》为线索,各课程间相辅相成,互相照应,系统地介绍了软件工程的整个学习过程。《数据结构应用设计》、《编译原理设计与实践》、《操作系统设计与实践》、《数据库系统设计与实践》等实践类教材以实验为主题,坚持理论内容以必须和够用为度,实验内容以新颖、实用为原则编写。通过一系列实验,培养学生的探究、分析问题的能力,激发学生的学习兴趣,充分调动学生的非智力因素,提高学生的实践能力。

相信本系列教材的出版,对于培养软件工程人才、推动我国计算机软件工程事业的发展必将起到积极作用。



2011 年 7 月

◎ 前言

Preface

算法的设计与分析是计算机科学的核心问题之一,也是计算机相关专业本科和研究生的一门重要的专业基础课。学生通过学习和研究各种计算机算法,掌握其算法分析和设计的主要思路,培养对算法复杂性的分析能力,为独立设计算法奠定坚实的理论基础,这对于每一位从事软件开发与研究的人来说都是非常重要和必不可少的。为了适应我国培养各类计算机人才的需要,本课程以算法设计策略为单元,系统地介绍计算机算法的设计方法和分析技巧,培养学生扎实的算法应用能力,并建立一定的理论基础。

全书共分7章,第1章介绍算法的基本概念,并对算法的计算复杂性和算法的描述作了简要阐述。然后围绕算法设计常用的基本设计策略组织了第2章至第7章的内容。

第2章介绍分治与递归,它是设计有效算法最常用的策略,也是必须掌握的方法。

第3章介绍贪心算法,它也是一种重要的算法设计策略,它与动态规划算法的设计思想有一定联系,但其效率更高。按贪心算法设计出的许多算法能导致最优解。其中有许多典型问题和典型算法可供学习和使用。

第4章是动态规划算法,以具体实例详述动态规划算法的设计思想、适用性以及算法的设计要点。

第5章是搜索算法,分别介绍回溯法和分支限界法。这章所介绍的算法适合于处理难解问题,两种解题思想各具特色,需要认真学习和掌握。

第6章介绍实用性很强的网络流与匹配算法。许多实际应用问题可以转化为网络流问题,特别是匹配问题在许多场合下都有重要应用。

第7章介绍线性规划算法。由于其在现代管理中经常采用,具有很强的实用性和工程性,所以越来越受到重视。

在本书各章的论述中,首先分析一种算法设计策略的基本思想,然后从解决实际问题入手,由易到难地描述几个经典的实例,使读者既能学到一些常用的精巧算法,又能通过对算法设计策略的反复应用,牢固掌握这些算法设计的基本思想,以便收到融会贯通之效。同时,本书特别注重对解同一个问题的不同算法的比较,使读者更容易体会到每一种具体算法的设计要点和各自的优缺点。

本书采用面向对象的C++语言作为算法描述手段,在保持C++优点的同时,尽量使算法描述简明、清晰。

本书由东北石油大学许少华任主编,东北石油大学富宇、哈尔滨工程大学潘海为、东北林业大学邱兆文任副主编,全书由富宇统稿。具体编写分工如下:第1章、第3章由许少华编写,第2章、第7章由潘海为编写,第4章、第5章由富宇编写,第6章由邱兆文编写。由于作者水平有限,书中难免存在疏漏与不足,敬请读者指正。

编者
2011年5月

◎ 目 录

Contents

第 1 章 算法概述	1
1.1 算法的概念	1
1.1.1 算法与程序	1
1.1.2 算法与数据结构	2
1.1.3 算法表示的基本方法	3
1.1.4 算法设计	6
1.2 算法复杂性分析的方法	7
1.2.1 两个算法的效率对比	8
1.2.2 算法复杂性的度量	10
1.2.3 复杂性的渐近性态及其阶	11
1.2.4 复杂性渐近阶的重要性	14
1.2.5 递归方程解的渐近阶的求法	16
小结	19
习题	20
第 2 章 分治与递归	22
2.1 递归概述	22
2.2 分治法概述	25
2.3 分治法的应用	26
2.3.1 排队购票问题	26
2.3.2 整数划分问题	28
2.3.3 “放苹果”问题	29
2.3.4 第 k 选择问题	30
2.4 典型问题分析	34
2.4.1 红与黑	34
2.4.2 循环赛日程表	35
2.4.3 0/1 背包问题	36
2.5 递归和递推	41
2.5.1 递归和递推的比较	41
2.5.2 “最少汽油过沙漠”问题	42

2.5.3 整数划分递推设计·····	44
2.6 递归的消除·····	48
小结·····	50
习题·····	50
第3章 贪心算法 ·····	52
3.1 贪心算法概述·····	53
3.2 活动安排问题·····	54
3.3 背包问题·····	57
3.4 贪心算法的两个基本要素·····	59
3.4.1 贪心选择性质·····	59
3.4.2 最优子结构性质·····	60
3.5 典型问题分析·····	61
3.5.1 最优装载问题·····	61
3.5.2 删数问题·····	62
3.5.3 均分纸牌·····	63
3.5.4 拦截导弹问题·····	64
小结·····	66
习题·····	66
第4章 动态规划 ·····	68
4.1 矩阵连乘问题·····	69
4.2 凸多边形最优三角剖分·····	73
4.3 最长公共子序列·····	75
4.4 DNA 序列比对·····	79
4.5 0/1 背包问题·····	81
4.6 多段图问题·····	83
4.7 数字三角形问题·····	85
4.8 备忘录方法·····	89
4.9 典型问题分析·····	90
4.9.1 游船费问题·····	90
4.9.2 航线设置·····	93
4.9.3 复制书稿·····	95
4.9.4 括号序列·····	97
小结·····	100
习题·····	100
第5章 搜索算法 ·····	102
5.1 问题解空间的分析·····	102

5.1.1 两种常见的树形问题解空间	102
5.1.2 解空间的两种搜索方式	103
5.2 回溯法概述	104
5.3 分支限界法概述	106
5.4 装载问题	108
5.4.1 回溯法解装载问题	108
5.4.2 分支限界法解装载问题	112
5.5 0/1 背包问题	116
5.5.1 回溯法解 0/1 背包问题	117
5.5.2 分支限界法解 0/1 背包问题	119
5.6 旅行售货员问题	124
5.6.1 回溯法解旅行售货员问题	124
5.6.2 分支限界法解旅行售货员问题	126
5.7 典型问题分析	129
5.7.1 子集和问题	129
5.7.2 油田勘探问题	131
小结	134
习题	134
第 6 章 网络流和匹配	135
6.1 最大流问题	135
6.1.1 概念	135
6.1.2 基本定理	138
6.1.3 求最大流的标号法	138
6.2 最小费用流问题	145
6.3 匹配	150
6.3.1 二部图	150
6.3.2 匹配	152
6.4 典型问题分析	155
6.4.1 物流运输	155
6.4.2 电力网络	158
6.4.3 选择课程	160
6.4.4 小行星	162
小结	163
习题	163
第 7 章 线性规划	166
7.1 线性规划问题及其表示	166

7.2 线性规划问题的数学形式	167
7.3 一般问题转化为约束标准型	169
7.4 线性规划的基、基础可行解	170
7.5 单纯形法算法	173
7.5.1 用消元法描述单纯形法算法	173
7.5.2 单纯形算法的框架	176
7.6 线性规划应用	180
小结	182
习题	182
参考文献	184

学习目标:理解算法的基本概念;理解算法、程序和数据结构的区别和联系;掌握算法表示的基本方法;理解算法复杂性概念;掌握算法复杂性的渐近性态的数学表述;理解递归方程解的渐近阶的求法。

1.1 算法的概念

1.1.1 算法与程序

在古希腊,欧几里得描述了求解两个整数的最大公因子的算法。要求解的问题描述为:“给定两个正整数 m 和 n ,求能同时整除 m 和 n 的最大整数,即它们的最大公因子。”欧几里得对解题步骤作出如下描述:

- ① 以 n 除 m ,并令所得余数为 r (必有 $r < n$)。
- ② 若 $r=0$,输出结果 n ,算法结束;否则继续步骤③。
- ③ 令 $m=n$ 和 $n=r$,返回步骤①继续进行。

从上述描述可以看出,算法就是求解问题的方法和步骤。这里的方法和步骤是一组严格定义了运算顺序的规则,每一个规则都是有效的、明确的,按此顺序将在有限次数下运算终止并获得问题的答案。

在计算机领域,算法的内涵要更丰富一些。为了使计算机能完成人们要求的工作,首先必须为如何完成预定的工作设计一个算法,然后再根据算法编写程序。计算机程序要对问题的每个对象和处理规则给出正确详尽的描述,其中程序的数据结构和变量用来描述问题的对象,程序结构、函数和语句用来描述问题的算法。算法是问题求解过程的精确描述,一个算法由有限个可完全机械地执行的、有确定结果的指令组成,指令正确地描述了的要完成的任务和它们被执行的顺序。计算机按算法指令所描述的顺序执行,算法的指令能在有限的步骤内终止——或终止于给出问题的解,或终止于指出问题对此输入数据无解。

虽然人们对算法一词非常熟悉,可到目前为止,对于算法尚没有统一而精确的定义。最为著名的定义是计算机科学家 D. E. Kunth 在其巨著《计算机程序的艺术》第一卷中所做的有关描述:一个算法,就是一个有穷规则的集合,其中之规则定义了一个解决某一特定类型问题的运算序列。Thomas H. Cormen 等人在《Introduction to Algorithms》一书中将算法描述为:算法是任何定义好了的计算过程,它取某些值或值的集合作为输入,并产生某些值或值的集合作为输出。综上所述,算法是将输入转化为输出的一系列计算步骤,是由基本运算及规定的运算顺序所构成的完整的解题步骤。或者看成按照要求设计好的、有限的、确切的计

算序列,并且这样的步骤和序列可以解决某一类问题,能够对特定规范的输入,在有限时间内获得所要求的输出。如果一个算法有缺陷,或者不适用于某个问题,那么执行这个算法将不会解决这个问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。

概括起来,一个算法应该具有以下五个重要的特征。

①有穷性:一个算法必须保证执行有限步之后结束。

②确定性:算法的每一个步骤必须有确切的定义。

③输入:一个算法有0个或多个输入,以描述运算对象的初始情况,所谓0个输入是指算法本身给定了初始条件。

④输出:一个算法有一个或多个输出,以反映对输入数据处理后的结果,没有输出的算法是毫无意义的。

⑤可行性:算法原则上能够精确地运行,人们可以用笔和纸做有限次运算后完成相同的工作。

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结构,它与算法设计是紧密相关的。

计算机科学家尼克劳斯·沃思(Niklaus Wirth)曾经做过这样的表述:程序=算法+数据结构,较为形象地说明了三者的关系。程序是算法的计算机高级语言描述,算法是程序的灵魂。研究和讨论算法是为了设计出更好的程序,设计好的算法都要转化为某种语言描述的程序才能在计算机上运行;或者说程序是算法表示的最终形态,程序只有装入计算机中运行(即程序的执行)时才能够起到对实际问题求解的作用。

1.1.2 算法与数据结构

计算机解决一个具体问题时,大致需要经过下列几个步骤:首先要从具体问题中抽象出一个适当的数学模型,然后设计一个解此数学模型的算法,最后编出程序、进行测试、调整直至得到最终解答。寻求数学模型的实质是分析问题,从中提取操作的对象,并找出这些操作对象之间含有的关系,然后用数学的语言加以描述。

数据结构主要研究数据的各种逻辑结构和存储结构,以及对数据的各种操作,因此,主要有三个方面的内容:数据的逻辑结构;数据的物理存储结构;对数据的操作(或算法)。数据结构是通过对数据的抽象与转换,帮助人们从生活里的具体事物中抽象出数学模型,进而写出适当的算法。

算法设计主要介绍计算机科学及应用领域常见的有代表性的非数值算法及算法设计的若干重要方法,阐述计算模型和时间复杂性的定义,介绍算法设计的几类方法,如分治法、回溯法、贪心法、动态规划法、分支限界法等,并结合某些有实用意义的经典算法来进行设计方法的探讨,介绍相关的复杂性理论。

算法设计和数据结构有很多相似的地方,它们都是对数据信息进行抽象,从中得到数学模型。虽然两者都是对数据的研究分析,但两者研究的侧重点不同。数据结构主要是对数据本身的抽象,是对相互之间存在一种或多种特定关系的数据元素集合的研究。通常情况下,精心选择的数据结构可以带来更高的运行或者存储效率。算法设计研究重点在于如何利用现在的数据结构写出更优秀的算法,如何得到更优秀的数据模型。

计算机算法与数据的结构密切相关,算法无不依附于具体的数据结构,数据结构直接关系到算法的选择和效率。通常,算法的设计取决于数据的逻辑结构,算法的实现取决于数据

的物理存储结构。

1.1.3 算法表示的基本方法

算法可以使用自然语言、伪代码、流程图和计算机程序设计语言等多种不同的方法来描述,不同的方法均有各自的优缺点。

1. 自然语言

所谓的“自然语言”指的是日常生活中使用的语言,如汉语、英语或数学语言。使用自然语言描述算法,人们比较容易接受和理解,如前面的欧几里得算法就是用自然语言描述的。

假设想计算1到 n 的累加和,为简单起见,设 n 的值不大于1 000。算法可以使用自然语言描述如下:

S1:输入 n (要求 $n \leq 1000$)

S2:累加和 sum 置初值0

S3:自然数 i 置初值1

S4:若 $i \leq n$,则重复执行

S41: $i+sum \rightarrow sum$

S42: $i+1 \rightarrow i$

S5:输出 sum ,结束

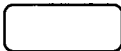
用自然语言描述算法具有许多缺点,一定要引起注意:

- ① 由于自然语言的歧义性容易导致算法描述的不精确。
- ② 自然语言的语句一般较长,从而导致了用自然语言描述的算法较长。
- ③ 自然语言的表示形式是顺序的,描述分支选择和转移等结构时不够直观。
- ④ 自然语言表示的算法不便于翻译成程序设计语言。

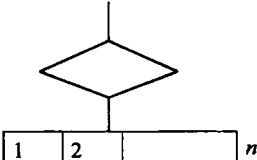
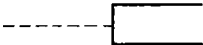
2. 算法流程图

流程图是用各种几何图形、流程线及文字说明来描述计算过程的框图。同自然语言相比,用流程图描述算法更为直观,可以一目了然;算法步骤间用流程线连接,次序关系清楚,容易理解,可以很方便地表示顺序、选择和循环结构,不依赖于任何计算机和程序设计语言,有利于在不同环境下的程序设计。下面的表1.1给出了流程图的常用符号。

表 1.1 流程图的常用符号

流程图符号	含 义
	数据输入/输出框,用于表示数据的输入和输出
	处理框,描述基本的操作功能,如“赋值”操作、数学运算等
	两分支判断框,根据框中给定的条件是否满足,选择执行两条路径中的一条
	开始/结束框,用于表示算法的开始和结束

续表 1.1

流程图符号	含 义
	连接符,用于连接流程图中不同地方的流程线
	流程线,表示流程的路径和方向
	多分支判断框,根据框中的“条件值”,选择执行多条路径中的一条
	注释框,框中内容是对某部分流程图的解释说明

欧几里得算法的流程图描述如图 1.1 所示。

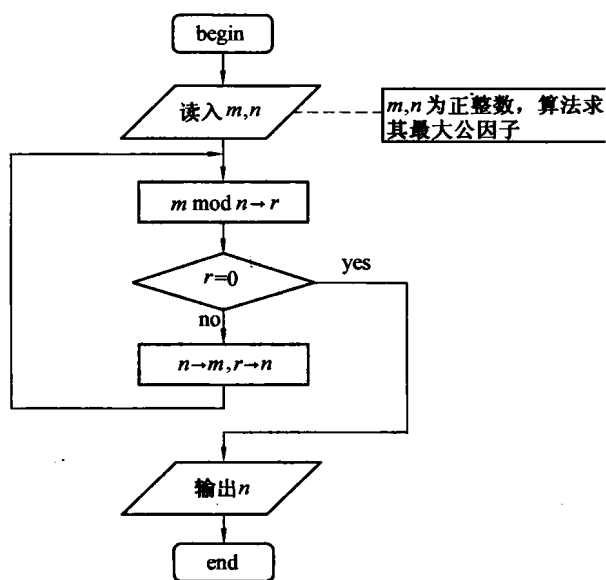


图 1.1 欧几里得算法的流程图表示

流程图也存在一些缺点,限制了它的使用:

- ① 流程图不是逐步求精的好工具,它会使人们过早地考虑算法的控制流程,而不去考虑算法的全局结构。
- ② 流程图的层次感不明显。
- ③ 不易于表示数据结构和模块调用关系等重要信息。
- ④ 用流程线代表控制流,控制转移随意性较大。若对流程线的使用不加限制,随着求解问题规模和复杂度的增加,流程图会变得很复杂,使人难以阅读、理解和修改,从而使算法的可靠性难以保证。

3. N-S 结构化流程图

为了克服传统流程图的缺点,美国学者纳斯(I. Nassi)和施内德曼(B. Shneiderman)提出了一种表示算法的较好工具——N-S 结构化流程图。N-S 图的主要特点是取消了流程线,全部算法由一些基本的矩形顺序排列组成一个大矩形框图,这种方式限制了程序的任意转移,而只能顺序执行,从而保证程序结构化的要求。图 1.2 给出了 N-S 图三种基本结构的表示方法。

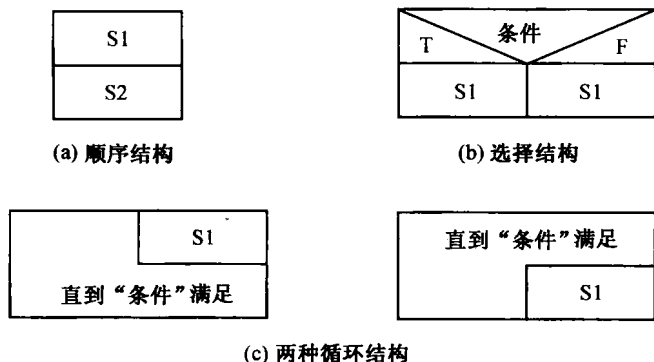


图 1.2 N-S 图的三种基本结构

欧几里得算法的 N-S 图描述如图 1.3 所示。

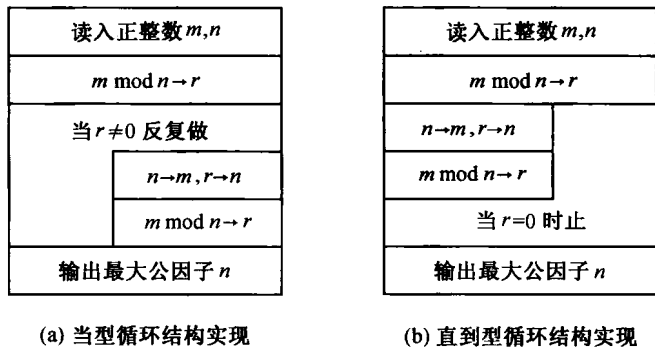


图 1.3 欧几里得算法的 N-S 图表示

N-S 图在一定程度上克服了传统流程图在结构化和层次感方面的缺点,但仍不易表示复杂的数据结构。

4. 伪代码

伪代码是介于自然语言和程序设计语言之间的一种描述方法,它在总体上采用某一程序设计语言的基本语法,但其中的具体处理步骤可以结合自然语言来说明。用伪代码写算法就如同写文章自上而下书写,每行写一个基本处理步骤,而用若干行写出一个基本模块,因而书写方便、格式紧凑、清晰易读,也容易转化为某一程序语言表示的程序。伪代码的优点是表达能力强、抽象性强,和图形工具相比较,便于修改,但直观性较弱。

下面给出欧几里得算法的一种伪代码描述:

Begin(算法开始)

```

Read(m,n)
m mod n → r
while r ≠ 0 do
{
    n → m
    r → n
    m mod n → r
}
write(n)
End(算法结束)

```

5. 程序设计语言

程序设计语言是计算机能够接受、理解和执行的算法描述工具。计算机不能直接识别自然语言、流程图、N-S图和伪代码等工具描述的算法,而设计算法的目的就是要用计算机来解决问题,所以算法最终都要用某一具体的程序设计语言来表示。从这个意义上讲,流程图、N-S图和伪代码都仅仅是为了求解问题而设计算法时的辅助工具。

和自然语言一样,程序设计语言也是串行的描述,很不直观。对于较复杂的问题,人们很难用程序设计语言直接写出程序。使用程序设计语言进行算法描述具有如下缺点:

① 与自然语言一样,程序设计语言也是串行的,算法的基本逻辑流程难以遵循,当算法较为复杂时这个问题就变得更加严重。

② 考虑到程序设计语言的不断更新,特定程序设计语言编写的算法限制了与他人的交流,不利于问题的解决。

③ 要花费大量的时间去熟悉和掌握某种特定的程序设计语言。

④ 要求描述计算步骤的细节而易忽视算法的本质。

⑤ 需要考虑语法细节,而扰乱算法设计的思路。

作为例子,下面给出欧几里得算法的 C++语言描述:

```

#include <stream. h>
main()
{
    int m,n,r;
    cout<<"请输入两个正整数:";
    cin>>m>>n;
    cout<<"\n"<< m<<"和"<<n<<"的最大公约数是:";
    r=m%n;
    while(r!=0)
    {
        m=n; n=r;
        r=m%n;
    }
    cout<<n<<"\n";
}

```

由于上述算法表示的方法各有优缺点,所以在算法设计阶段,一般是先采用某个辅助工具来描述算法,在算法设计好之后,再把它转化为某一具体程序设计语言描述的程序。

1.1.4 算法设计

算法设计是用计算机解决问题的一个重要步骤,其任务是对各类具体问题设计出良好