



高等院校实验教材 信息技术类

数据结构与算法 实验教程

吴艳 周苏 李益明 柳俊 等 编著



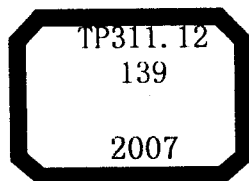
科学出版社
www.sciencep.com

清华大学出版社 清华大学出版社

数据结构与算法 实验教程

王磊 何磊 李国明 李国平 编著

清华大学出版社



高等院校实验教材 信息技术类

数据结构与算法实验教程

吴 艳 周 苏 李益明 柳 俊 等 编著

科学出版社

北 京

内 容 简 介

本书是为“数据结构”课程编写的以实验为主体开展教学的教材,也可作为课程的实验辅助教材。全书采用 C 语言作为数据结构与算法的描述语言,通过一系列实验练习,把数据结构的概念、理论知识与技术融入到实际应用中去,从而加深对本课程的认识和理解,逐步熟悉和掌握自顶向下的结构化程序设计方法,提高编程能力和综合分析能力,并为今后学习面向对象程序设计做一些铺垫。本书实验内容涉及数据结构和算法分析基础、线性表、栈和队列、串、树和二叉树、图,以及查找与内部排序等“数据结构”课程的各个方面。每个实验都包含背景知识介绍和按步骤进行的实验指导等,实验内容的组织充分顾及了不同的难易程度,富有挑战性。

本书可作为高等院校计算机、信息管理专业的主教材或实验辅助教材。

图书在版编目(CIP)数据

数据结构与算法实验教程/吴艳等编著. —北京:科学出版社,2007

(高等院校实验教材 信息技术类)

ISBN 978-7-03-018490-0

I.数… II.吴… III. ①数据结构—高等学校—教材②算法分析—高等学校—教材 IV.TP311.12

中国版本图书馆 CIP 数据核字(2007)第 012468 号

责任编辑:陈晓萍/责任校对:赵 燕

责任印制:吕春珉/封面设计:三函设计

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

铭浩彩色印装有限公司印刷

科学出版社发行 各地新华书店经销

*

2007 年 2 月第 一 版 开本: B5 (720×1000)

2007 年 2 月第一次印刷 印张: 20

印数: 1—3 000 字数: 392 000

定价: 26.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

销售部电话 010-62136131 编辑部电话 010-62138978-8003

前 言

高等教育的大众化对信息技术类专业课程的教学提出了更高的要求，新的高等教育形势需要我们积极研究新的教学方法。在长期的教学实践中我们体会到，“因材施教”，把实验实践环节与理论教学相融合，抓实验实践教学促进学科理论知识的学习，是有效地提高教学效果和教学水平的重要方法之一。为了完善相关课程的实验教学，为社会培养合格和适用的专业人才，迫切需要优秀的系列实验教材。

为此，我们邀请长期工作在教学第一线且年富力强的各课程资深专业教师，在实验内容选择、实验步骤设计和实验文档组织等诸方面都做了精心的考虑和安排，编写了这套“高等院校实验教材（信息技术类）”。首批实验教材所涉及的内容包括：操作系统原理、软件工程、多媒体技术、网页设计与网站建设、信息资源管理、电子商务概论、汇编语言程序设计、动态网页技术、数据库技术、数据结构与算法、信息安全技术、面向对象程序设计等专业课程，以及 Visual C++、Java、Delphi 等程序设计课程。

作为与课程主教材配套的实验教材，本套教材的编写原则是：依据课程教学大纲，充分理解课程的大多数主教材，遵循教学的规律和节奏，充分体现实验的可操作性，可以与课程主教材辅助配套使用。另一方面，由于本套教材的实验设计中包含了必要的理论知识，所以，在主要强调实践的课程中，也可作为主教材被选用，同时还是良好的自学教材。编写本套教材的目的旨在很好地推动课程的教学发展，辅助老师教，帮助学生学，从而帮助读者切实把握本学科的知识内涵和理论与实践的水平。

全书采用 C 语言作为数据结构与算法的描述语言，通过一系列实验练习，把数据结构的概念、理论知识与技术融入到实际应用中去，从而加深对本课程的认识和理解，逐步熟悉和掌握自顶向下的结构化程序设计方法，提高编程能力和综合分析能力，并为今后学习面向对象程序设计做一些铺垫。

本书实验内容涉及数据结构和算法分析基础、线性表、栈和队列、串、树和二叉树、图，以及查找与内部排序等“数据结构”的各个方面。全书包括 16 个实验练习、1 个实验总结和 1 个附录实验。每个实验都包含背景知识介绍和按步骤进行的实验指导等，实验内容的组织充分顾及了不同的难易程度，富有挑战性。每个实验均留有“实验总结”和“教师评价”部分；每个单元设计了“实验单元的学习评价”；全部实验完成之后还设计了“课程学习能力测评”等内容。希望以此方便师生交流对学科知识、实验内容的理解与体会，方便老师对学生学习情况的评估。

赵端阳、吴林华等同志参加了本书的部分编写工作，本书的编写得到了浙江工业大学之江学院、浙江大学城市学院、温州大学、浙江商业职业技术学院等多所院校领导及师生的大力支持，在此一并表示感谢！

本书相关的实验素材可从科学出版社网站 (www.sciencep.com) 的下载区下载。欢迎教师与作者交流并索取为教学配套的相关资料 (E-mail: zs@mail.hz.zj.cn 或 QQ: 81505050)。

编者

2006年11月

于西子湖畔

读者指南

任何实际问题只有建立了数学模型才可以被计算机计算，而数据结构就是实际问题中元素的数学抽象，算法则是建立和解决数学模型的方法。

“数据结构”这个术语在整个计算机科学与技术领域中被广泛使用，它被用来反映一个数据的内部构成，即一个数据由哪些成分数据构成，以什么方式构成，呈什么样的结构等。数据结构是数据存在的形式，也是信息的一种组织方式，其目的是为了提高算法的效率，它通常与一组算法的集合相对应，通过这组算法集合可以对数据结构中的数据进行某种操作。

“数据结构”是计算机及其相关专业的一门核心课程，具有承上启下的地位和作用，“程序设计语言”（例如 C 或 C++）和“离散数学”是它的先导课程，“操作系统”、“数据库原理”、“软件工程”等是它的后续课程。

全书采用 C 语言作为数据结构与算法的描述语言，通过一系列实验练习，把数据结构的概念、理论知识与技术融入到实际应用中去，从而加深对本课程的认识和理解，逐步熟悉和掌握自顶向下的结构化程序设计方法，提高编程能力和综合分析能力，并为今后学习面向对象程序设计做一些铺垫。

本书中的所有程序都通过了 Borland C++ 5.0 和 Microsoft Visual C++ 6.0 软件开发环境下的调试运行，以尽可能地保证所给出算法和程序的正确性和有效性。

读者对象

高等院校计算机和信息管理等相关专业“数据结构”课程的学生可把此书作为学习的实验辅助教材；由于本书在实验设计中包含了必要的理论知识，所以，在主要强调实践的課程中，本书也可以作为主教材使用。对于已经具备初步的计算机应用和程序设计知识，并希望通过进一步学习得到提高、希望通过计算机等级考试的读者来说，本书也是一本自学和继续教育的良好读物。

相信本书的实验内容有助于“数据结构”课程的教与学，有助于读者对掌握和理解本课程内容建立起足够的信心和兴趣。

实验内容

本书的实验练习几乎覆盖了“数据结构”课程教学的各个方面，内容涉及数据结构和算法分析基础、线性表、栈和队列、串、树和二叉树、图，以及查找与

内部排序等,全书包括 16 个实验练习、1 个实验总结和 1 个附录实验。每个实验都包含背景知识介绍和实验指导等,实验练习的难易程度不同,以帮助读者加深对教材中概念的理解,并逐步熟悉和掌握自顶向下的结构化程序设计方法。

实验 1: 数据结构和算法分析基础。包括数据结构和算法的计算环境、抽象数据类型的表示和实现、算法和算法分析等实验。通过实验来理解抽象数据类型的特点、定义方法和在 C 语言环境下实现的方法;掌握算法的主要特征和描述方法;尝试通过具体的算法结构,计算算法的时间复杂度和空间复杂度,并对算法进行定性或定量评价。

实验 2: 线性表。包括线性表的顺序表示和实现、线性表的链式表示和实现等实验。通过实验,掌握顺序表和链表的存储结构定义;实现基本操作算法的描述和分析;进一步理解实现复杂操作的 C 语言程序的结构,从而更深入理解结构化程序设计的方法;与此同时,加深理解 C 语言中函数和结构体的结构、语法和使用方法。

实验 3: 栈和队列。包括栈、队列和递归算法等实验。通过实验,掌握顺序栈和链式栈、顺序队列和链式队列的存储结构定义;掌握栈和队列基本操作的过程及实现的方法;理解递归算法的思路、使用条件以及设计方法,并且了解由递归算法到非递归算法的转换过程中栈的作用;通过实例,进一步加深对栈和队列特点的理解,并了解这两种结构在解决实际问题时的区别,从而更合理地选择适当的数据结构解决实际应用。

实验 4: 串。包括串的 3 种常见的存储结构定义、在具体定义下串的基本操作和实现。通过这个实验,加深理解串这种常见的数据结构的特点和存储结构定义方法;掌握串的基本操作算法的描述,并根据算法的结构评价算法;理解串操作的实现方法,理解一个合理的存储结构定义对具体操作实现的重要性,并注重提高算法的健壮性,从而更好地理解一个好算法所需要的各种综合因素。

实验 5: 树和二叉树。包括二叉树及其基本操作、哈夫曼树和哈夫曼编码等实验。通过实验,掌握二叉树的不同存储结构定义,并理解在具体的应用中采用合理的存储结构的思路;加深理解递归算法的设计思路,体会栈在递归算法中的作用;理解广度搜索算法的设计思路,体会队列在实际应用中的作用;理解哈夫曼树的特点和存储结构的定义思路、哈夫曼树建立的算法和实现,并在此基础上体验哈夫曼编码的特点和设计过程,弄清实现哈夫曼编码程序的结构,从而加深对 C 语言中字符串和指针的理解并灵活运用。

实验 6: 图。包括图的表示和实现、图的应用、贪心算法等实验。通过实验,掌握图的各种存储结构的定义方法,从而更进一步理解如何根据实际问题设计合理的数据结构;掌握图的两种遍历方法,即图的深度优先搜索和广度优先搜索,并根据这两种操作的实现过程确定算法中所需要采用的辅助数据结构;能根据实

实际问题选择图的类型,并采用合理的存储结构,理解解决最小生成树、拓扑排序、关键路径和最短路径等实际应用的算法描述,并能根据算法写出相应的程序代码。根据贪心算法的核心思想理解贪心算法的特点,并学会设计简单的贪心算法。

实验 7: 查找与内部排序。包括查找、内部排序等实验。通过实验,掌握查找表的特点,并能根据具体的查找方法定义合理的查找表的存储结构;掌握各种查找表的实现思路,能根据查找思路写出算法的描述,并对算法进行评价;掌握哈希查找的特点和影响哈希查找的因素,从而构造合理的哈希函数、采用适当的解决冲突的方法,加深对 C 语言中指向函数的指针的理解。通过实验,熟悉各种排序方法的特点,并能根据具体的实现方法定义合理的待排序数据的存储结构;掌握各种排序方法的实现思路,能根据排序思路写出算法的描述,并对相应算法进行时间复杂度和空间复杂度计算;体会各种排序算法在最好情况下和最坏情况下的算法评价,并能根据待排序数据的实际分布,制订相应的排序方法。

附录: 算法描述绘图工具 Visio 实验。学会在 Visio 环境下制作算法的程序流程图和 N-S 图等。

各个实验练习之间的难度不断增加,从这个意义上讲,应该在完成前面部分的练习之后再进行后面实验的练习。如果在做练习时遇到了困难,你可能需要查阅更早的练习来帮助解决问题。

尽管在各个实验练习中包含了对关键概念和知识的简要介绍,但如果在做这些实验操作练习的同时,学习相关的“数据结构”和“C 语言程序设计”等课程,则可以从实验和练习中获取更多的知识,来提高自己的综合分析能力和编程能力。

实 验 要 求

根据不同的教学安排和要求,“数据结构与算法”的实验学时数也有所不同。

致教师

数据结构与算法的应用面广,涉及技术领域宽泛,也被人们赋予了很高的期望值。另一方面,要学好“数据结构”课程,仅仅通过课堂教学或自学获取理论知识是远远不够的。因此,要让学生真正理解数据结构与算法的基础理论知识,具备将数据结构与算法知识应用于社会实践的能力,积极加强数据结构课程的实验环节是至关重要的。

作为一本“数据结构”课程的以实验为主体开展教学的教材,本书结合软件图形描述工具和相应的高级语言开发工具,通过提供一组与单元知识密切相关的实验练习作为对数据结构主教材的补充,有助于学生对理论知识的理解,有助于

提高学生的应用和开发能力。

为方便教师对课程实验的组织，我们在实验内容的选择、实验步骤的设计和实验文档的组织等诸方面都做了精心的考虑和安排。任课教师不需要投入很多精力来设计实验练习。相反，教师和学生都可以通过本书提供的实验练习来理解概念和实现应用。任课教师需要在开头的几个实验中引导学生熟悉实践环境、理解抽象的数据类型的定义方法和体会自顶向下的结构化程序设计风格，使学生能按照实验步骤循序渐进地进行数据结构设计、算法设计和程序的调试。

本书的全部实验都经过了教学实践的检验，取得了较好的教学效果，大部分学生在这种实验模式下进行实践后，分析能力和编程能力得到了显著提高。根据经验，虽然大部分的实验确实能够在一次实验课的时间内完成，但学生中普遍存在几个问题：

1) 实验前的准备工作不充分。学生常常会忽视对每个实验的“背景知识”的阅读和理解，而一味只求完成实验步骤。

2) 实验中不注重理解所做的实验内容。只是按部就班地照着实验步骤进行程序代码的输入和调试，遇到错误提示，不会积极主动地进行排错。

3) 实验步骤完成后不能及时进行总结。为了赶时间，往往草草了事，没有投入时间对刚完成的实验内容进行消化，往往需要反复多次才能熟悉原本较为容易掌握的知识内容。

因此，为了保证实验的质量，建议教师重视对教学实验环节的组织，例如：

1) 要求学生实验前对实验内容进行预习，并把预习重点放在实验的“背景知识”部分。实验指导老师在实验开始初期对学生的预习情况进行检查，计入实验成绩。

2) 明确要求学生重视对实验内容的体会和理解，认真完成“实验总结”，并把这部分内容作为实验成绩的主要评价成分，以促使学生对所学知识的理解，并做到举一反三。

3) 对于有条件的学校（例如，学生自备电脑，或实验室有足够的课余上机时间安排），不理解的实验部分可以建议学生反复实践，以加深对核心和重要内容的理解。

本书提供了足够数目的实验练习，一些对学习和实验来说较难的内容我们在该节的名称前面加注了星号（“*”），例如实验 2.2 中的“约瑟夫环问题的实现”，实验 3.1 中的“括号匹配的检验”、“表达式求解”、“迷宫求解”，实验 3.2 中的“农夫过河”、“停车场管理”以及实验 6.3 的“贪心算法”等内容。这部分内容以读懂为要求，对学生的理解和编程有一定指导意义，请教师酌情选用。

我们希望有机会不断更新本书的版本，使教师有选择新的、更好的实验的机会。

如果需要，教师可以根据学生的学习情况，在现有实验的基础上，结合当前

热门的实际应用给出一些要求、指导和布置,以进一步发挥学生的潜能和激发学生学习的主动性和创造性。

每个实验均留有“实验总结”和“教师评价”部分;每个单元设计了“实验单元的学习评价”;全部实验完成之后的实验总结部分还设计了“课程学习能力测评”等内容,书后则提供了“实验成绩记录”。希望以此方便师生交流对学科知识、实验内容的理解与体会,方便老师对学生实验成绩的记录与管理,及时发现学生在学习过程中的问题,并加以针对性的指导,以及对学生学习情况进行必要的评估。

致学生

对于计算机及其相关专业的学生来说,数据结构肯定是需要掌握的重要专业基础知识之一。但是,单凭课堂教学和一般作业,要真正领会数据结构课程所介绍的概念、原理、方法和技巧等,是很困难的,应该通过大量的上机实验,才能真正理解相关知识,并做到灵活运用。

另一方面,经验表明,学习尤其是真正体会和掌握数据结构与算法的最好方式是对它进行充分的实践,无疑,通过了解、熟悉和掌握数据结构与算法程序设计,是应用数据结构与算法知识的重要途径。学生必须通过了解、熟悉一些经典算法,来逐步掌握一些编程技巧,提高自己的分析和判断能力。

以下几点对于提高实验效果是非常重要的:

1) 实验前必须做好预习工作。请务必预习每个实验的“背景知识”部分。“背景知识”是主教材内容的补充和延伸,和实验内容有着密切的关系,读者可以通过看教材、查阅相关文献或工具书来理解和丰富“背景知识”中叙述的知识内容。

2) 实验过程中必须认真理解、体会各个实验步骤。在输入程序代码时,必须认真理解程序的结构和编程方法;在调试程序时,注意出错的原因及修正的方法,坚持下来,就能减少错误率,提高调试程序的效率。

3) 实验完成后做好实验总结。必须认真填写每个实验后的“实验总结”,把对本次实验的认识、感受和意见等表达出来,同时,也可以提出实验中你还没有解决的问题,及时和教师进行交流,及时得到纠正。

本书所提供的实验毕竟有限,程度好的同学可以在本书实验的基础上,结合实际应用进行合理的数据结构设计和算法设计,以发挥自己的潜能和创造性。

实 验 设 备

个人计算机在学生,尤其是计算机及其相关专业学生中的普及,使得我们有

机会把实验任务分别利用课内和课外时间来完成，以获得更多的锻炼。这样，对实验室和个人计算机的配置就有不同的要求。

实验室设备与环境

用来进行本书数据结构实验的实验室环境，大都需要其计算机设备安装有 Borland C++ 5.0 或 Microsoft Visual C++ 6.0 开发环境。

由于部分实验有可能无法一次完成，有些实验在内容上有一定的互通性和连贯性，所以，实验室设备应能帮助并注意提醒学生妥善保存其实验内容。

个人实验设备与环境

用于本书数据结构实验的个人计算机环境，一般建议在其 Windows 2000 Professional、Windows XP Professional 操作系统中安装有 Borland C++ 5.0 或 Microsoft Visual C++ 6.0 开发环境。个人计算机环境需要为实验准备足够的硬盘存储空间，以方便实验软件的安装和实验数据的保存。

在利用个人计算机完成实验时，要重视在操作中系统所显示的提示甚至警告信息，注意保护自己的数据和计算环境的安全，做好必要的数据备份工作，以免产生不必要的损失。

由于有些实验在内容上有一定的互通性和连贯性，所以，要注意妥善保存自己的实验内容。

没有设备时如何使用本书

如果本书的读者由于某些客观原因无法获得必要的实验设备时，也不用失望，我们相信您仍将从本书中受益。全书以循序渐进的方式介绍了每个实验的背景知识和实验任务，其中也包含了相当一部分知识内容。读者通过认真阅读“背景知识”，仔细分析实验中给出的实例和程序代码，也能在一定程度上有所收获。

目 录

实验 1	数据结构和算法分析基础	1
1.1	数据结构与算法的计算环境	1
1.2	抽象数据类型的表示和实现	7
1.3	算法和算法分析	20
实验 2	线性表	28
2.1	线性表的顺序表示和实现	28
2.2	线性表的链式表示和实现	45
实验 3	栈和队列	67
3.1	栈	67
3.2	队列	89
3.3	递归算法	113
实验 4	串	124
实验 5	树和二叉树	144
5.1	二叉树及其基本操作	144
5.2	哈夫曼树和哈夫曼编码	169
实验 6	图	179
6.1	图的表示和实现	179
6.2	图的应用	201
* 6.3	贪心算法	219
实验 7	查找和内部排序	236
7.1	查找	236
7.2	内部排序	256
实验 8	数据结构与算法实验总结	281
8.1	实验的基本内容	281
8.2	实验的基本评价	283
8.3	课程学习能力测评	284
8.4	数据结构与算法实验总结	285
8.5	实验总结评价（教师）	286
附录	算法描述绘图工具 Visio 实验	287
实验成绩记录		303
主要参考文献		305



实验 1

数据结构和算法分析基础

1.1 数据结构与算法的计算环境

(实验估计时间: 90 分钟)

1.1.1 背景知识

除了进行科学计算之外, 计算机已经被广泛地应用在控制、管理和数据处理等非数值计算的领域中。与此相应, 处理对象也由早先纯粹的数值发展到字符、表格和图形图像等各种具有一定结构的数据, 这给计算机程序设计带来了新的问题。为了编写一个“好”的程序, 必须明确处理对象的特征及各对象之间的关系。这就是“数据结构”这门学科形成和发展的背景。

任何实际问题只有建立了数学模型才可以被计算机计算, 而数据结构就是实际问题中操作对象(元素)的数学抽象, 算法则是建立和解决数学模型的方法。

数据结构用来反映计算机加工处理的对象, 即数据的内部构成, 亦即数据由哪几部分构成, 以什么方式构成, 呈什么样的结构等。数据结构包括逻辑结构和物理结构。这里的逻辑结构和物理结构是指一个事物的两个方面, 而不是指两个不同的对象。逻辑结构反映数据元素之间的逻辑关系, 而物理结构反映了数据元素在计算机内部的存储安排, 也称为存储结构。

数据结构是数据存在的形式, 也是信息的一种组织方式, 其目的是为了提高算法的效率。数据结构通常与一组算法的集合相对应, 通过这组算法集合可以对数据结构中的数据进行某种操作。由于相同算法中的抽象数据类型用不同的数据结构来表示, 会造成不同的执行效率, 这就有必要来研究不同数据结构表示的效率差异及其适用场合。

1. 数据结构的研究对象

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3个方面的内容,即数据的逻辑结构、数据的存储(物理)结构和对数据的操作(或算法)等。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存储结构。

2. 数据结构的形式化定义

数据是指由有限的符号(比如“0”和“1”,具有其自己的结构、操作和相应的语义)组成的元素的集合。结构是元素之间关系的集合。通常来说,一个数据结构 DS 可以表示为一个二元组:

$$DS = (D, S)$$

其中, D 是数据元素的集合(或者是“结点”,可能还含有“数据项”或“数据域”), S 是定义在 D (或其他集合)上的关系的集合, $S = \{ R \mid R : D \times D \times \dots \}$, 称之为元素的逻辑结构。

例如,学生排队问题数据结构的二元组为

$$DS = (\text{student}, \text{order})$$

其中,

$$\text{student} = \{ a_i \mid a_i \in D, D \text{ 为同类学生的信息集} \}$$

$$\text{order} = \{ \langle a_i, a_{i+1} \rangle \mid a_i, a_{i+1} \in \text{student}, i = 1, 2, \dots, n-1 \}$$

(学生在队伍中的关系是前后次序关系)

逻辑结构有4种基本类型:集合结构、线性结构、树状结构和网络结构。表和树是最常用的两种高效数据结构,许多高效的算法都用这两种数据结构来设计和实现。表(全序关系)是线性结构的,树(偏序或层次关系)和图(局部有序)是非线性结构的。

数据结构的物理结构是指逻辑结构的存储镜像(image),即用怎样的物理存储方式来表示其逻辑关系。数据结构 DS 的物理结构 P 对应于从 DS 的数据元素到存储区 M (维护着逻辑结构 S) 的一个映射:

$$P : (D, S) \rightarrow M$$

1) 存储器模型。一个存储器 M 是一系列固定大小的存储单元,每个单元 U 有一个唯一的地址 A (U), 该地址被连续地编码。每个单元 U 有一个唯一的后继单元 $U' = \text{succ}(U)$ 。

P 有4种基本映射模型:顺序(sequential)、链接(linked)、索引(indexed)和散列(hashing)映射。因此,我们至少可以得到 4×4 种可能的物理结构(并不是所有的可能组合都合理):

sequential	(sets)
linked	lists
indexed	trees

hash

graphs

例如, 学生排队问题中, 因为数据元素之间是一种前后次序关系, 即采用的逻辑数据结构为线性结构, 在计算机中可以开辟一个连续的存储空间来存放数据元素, 故可采用顺序结构作为其物理结构。

2) 数据结构 DS 上的操作。所有定义在 DS 上的操作在改变数据元素 (结点) 或结点的域时必须保持 DS 的逻辑和物理结构。

3) DS 上的基本操作。任何其他对 DS 的高级操作都可以用这些基本操作来实现。最好将 DS 和它的所有基本操作看作为一个整体——称之为模块。我们可以进一步将该模块抽象为数据类型 (其中 DS 的存储结构被表示为私有成员, 基本操作被表示为公共方法)。例如, 学生排队问题中的基本操作有出列 (删除操作)、入列 (插入操作) 和点名 (检索) 等。

3. 算法

算法 (algorithm) 是在有限步骤内求解某一问题所使用的一组定义明确的规则。通俗地说, 就是计算机解题的过程。在这个过程中, 无论是形成解题思路还是编写程序, 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。

一个算法应该具有以下 5 个重要的特征。

- 1) 有穷性: 一个算法必须保证执行有限步骤之后结束。
- 2) 确切性: 算法的每一步骤必须有确切的定义。
- 3) 可行性: 算法中的每一步骤必须充分可及, 即算法原则上能够精确地运行, 而且人们用笔和纸做有限次运算后即可完成。
- 4) 输入: 一个算法有 0 个或多个输入, 以刻画运算对象的初始情况。所谓 0 个输入是指算法本身定义了初始条件。
- 5) 输出: 一个算法有一个或多个输出, 以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

4. 算法表达中抽象机制

要用计算机解决一个稍微复杂的实际问题, 大体都要经历如下步骤 (如图 1.1 所示)。

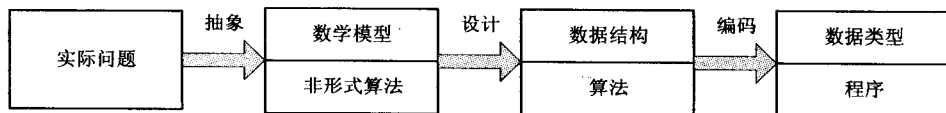


图 1.1 从问题到程序

1) 将实际问题数学化。即把实际问题抽象为一个带有一般性的数学问题。这一步要引入一些数学概念, 精确地阐述数学问题, 弄清问题的已知条件、所要求的结果, 以及在已知条件和所要求的结果之间存在着的隐式或显式的联系等。

2) 对于确定的数学问题, 设计其求解的方法, 即所谓的算法设计。这一步要建立问题的求解模型, 即确定问题的数据模型并在此模型上定义一组运算。然后, 借助于对这组运算的调用和控制, 从已知数据出发导向所要求的结果, 形成算法并用自然语言来表述。这种语言还不是程序设计语言, 不能被计算机所接受。

3) 用计算机的某种程序设计语言来表达已设计好的算法。换句话说, 将非形式自然语言表达的算法转变为一种程序设计语言表达的算法。这一步就叫程序设计。

4) 在计算机上编辑、调试和测试编制好的程序, 直到输出预期的结果。

在这里, 我们只关心第 3) 步, 而且把注意力集中在算法程序表达的抽象机制上, 目的是引入一个重要的概念——抽象数据类型, 同时为大型程序设计提供一种相应的自顶向下逐步求精、模块化的具体方法, 即运用抽象数据类型来描述程序的方法。

抽象数据类型 (abstract data types, 简称 ADT) 是算法设计和程序设计中的重要概念。严格地说, 它是算法的一个数据模型连同定义在该模型上、作为该算法构件的一组运算。这个概念明确地把数据模型与作用在该模型上的运算紧密地联系起来。

事实正是如此。一方面, 数据模型的运算依赖于数据模型的具体表示, 因为数据模型运算以数据模型中的数据变量作为运算对象, 或作为运算结果, 或二者兼而为之; 另一方面, 有了数据模型的具体表示以及数据模型运算的具体实现, 运算的效率随之确定。

于是, 就有这样一个问题: 如何选择数据模型的具体表示使该模型上的各种运算的效率都尽可能地高? 很明显, 对于不同的运算组, 为使组中所有运算的效率都尽可能地高, 其相应的数据模型所具体表示的选择将是不同的。在这个意义下, 数据模型的具体表示又反过来依赖于数据模型上定义的那些运算。特别是, 当不同运算的效率互相制约时, 还必须事先将所有运算的相应使用频度排序, 让所选择的数据模型的具体表示优先保证使用频度较高的运算有较高的效率。数据模型与定义在该模型上的运算之间存在着的这种密不可分的联系, 是抽象数据类型的概念产生的背景和依据。

一些基本抽象数据类型, 包括表、栈、队列、串、树、二叉树和图等, 是最基本和最简单的, 并且是实现其他抽象数据类型的基础。

高级抽象数据类型主要包括集合、字典、散列表、有序字典、并查集、优先队列、优先级树和堆、检索树、搜索树、分离集合等。