

面向计算机科学与技术专业规范系列教材



数据结构与算法

吴跃 主编

李树全 尚明生 陈端兵 参编

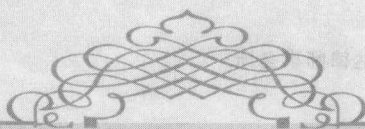


Data Structures & Algorithms



机械工业出版社
China Machine Press

面向计算机科学与技术专业规范系列教材



数据结构与算法

Data Structures & Algorithms



机械工业出版社
China Machine Press

本书以基本数据结构和算法设计策略为知识单元，系统地介绍了数据结构的知识与应用、计算机算法的设计与分析方法，主要内容包括线性表、树、图和广义表、算法设计策略以及查找与排序算法等。

本书注重理论与实践相结合，内容深入浅出，可以作为高等院校计算机学科相关专业的教材或参考书，同时对计算机科技工作者也有参考价值。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目 (CIP) 数据

数据结构与算法 / 吴跃主编. — 北京: 机械工业出版社, 2010.2

(面向计算机科学与技术专业规范系列教材)

ISBN 978-7-111-28825-1

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 (2009) 第 207700 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 迟振春

北京市荣盛彩色印刷有限公司印刷

2010 年 2 月第 1 版第 1 次印刷

185mm×260mm·16.25 印张

标准书号: ISBN 978-7-111-28825-1

定价: 28.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

出版者的话

机械工业出版社华章公司是国内重要的教育出版公司，培生教育集团（拥有 Addison Wesley、Prentice Hall 等品牌）是全球知名的教育出版集团，双方在过去长达十余年的合作中秉承“全球采集内容，服务教育事业”的理念，遴选、移译了国外大量的在计算机科学界享誉盛名的专家名著与名校教材，其中包括 Donald E. Knuth、Alfred V. Aho、Jeffrey D. Ullman、John E. Hopcroft、Dennis Ritchie 等大师名家的经典作品（收录在大理石封面的“计算机科学丛书”中），这些作品对国内计算机教育及科研事业的发展起到了积极的促进作用。

随着国内计算机科学与技术专业学科建设的不断完善、教学研究的蓬勃发展，以及教材改革的逐渐深化，计算机科学与技术专业的优秀课程及教材不仅仅是“引进来”（版权引进），而且需要“走出去”（版权输出）了。

近几年以来，教育部计算机科学与技术专业教学指导分委员会根据我国计算机专业教育的现状以及社会对人才的需求，发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》）。为配合《规范》的实施推广，同时为落实中央“提高高等教育质量”的最新指导思想，在教育部计算机科学与技术专业教学指导分委员会的指导下，在国内知名高校众多教授的帮助下，我们出版了这套“面向计算机科学与技术专业规范系列教材”。

本套教材的作者在长达数十年的科研和教学经历中积累了大量的知识和经验，也奠定了他们在学术和教学领域的地位，教材的内容体现了他们的教学思想和教学理念，本套教材也是传承他们优秀教学成果的最好载体，是中国版的专家名著和名校教材，相信它们的出版对提高计算机科学与技术专业的教育水平和教学质量能够起到积极的作用。

华章与培生作为专业的出版团队，愿与高等院校的老师共同携手，在这套教材的出版上引进国际先进教材出版经验，在教学配套资源的建设上做出新的尝试，为促进中国计算机科学与技术专业教育事业的发展，为增进中国与世界文化的交流而努力。



PEARSON
Education

华章教育 培生教育集团



序 言

近 20 年里, 计算机学科有了很大的发展, 人们普遍认为, “计算机科学”这个名字已经难以涵盖该学科的内容, 因此, 改称其为计算学科 (Computing Discipline)。在我国本科教育中, 1996 年以前曾经有计算机软件专业和计算机及应用专业, 之后被合并为计算机科学与技术专业。2004 年以来, 教育部计算机科学与技术专业教学指导分委员会根据我国计算机专业教育和计算学科的现状, 为更好地满足社会对计算机专业人才的需求, 发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范 (试行)》(以下简称《规范》), 提出在计算机科学与技术专业名称之下, 构建计算机科学、计算机工程、软件工程和信息技术四大专业方向。《规范》中四大专业方向的分类, 在于鼓励办学单位根据自己的情况设定不同的培养方案, 以培养更具针对性和特色的计算机专业人才。

为配合《规范》的实施, 落实中央“提高高等教育质量”的精神, 我们规划了“面向计算机科学与技术专业规范系列教材”。本系列教材面向全新的计算学科, 针对我国高等院校逐步向新的计算机科学与技术专业课程体系过渡的趋势编写, 在知识选择、内容组织和教学方法等方面满足《规范》的要求, 并与国际接轨。本套教材具有以下几个特点:

(1) **体现《规范》的基本思想, 满足其课程要求。**为使教材符合我国高等院校的教学实际, 编委会根据《规范》的要求规划本套教材, 广泛征集在国内知名高校中从事一线教学和科研工作、经验丰富的优秀教师承担编写任务。

(2) **围绕“提高教育质量”的宗旨开发教材。**为了确保“精品”, 本系列教材的出版不走盲目扩大的路子, 每本教材的选题都将由编委会集体论证, 并由一名编委担任责任编委, 最大程度地保证这套教材的编写水准和出版质量。

(3) **教材内容的组织科学、合理, 体系得当。**本套教材的编写注重研究学科的新发展和新成果, 能够根据不同类型人才培养需求, 合理地进行内容取舍、组织和叙述, 还精心设计了配套的实验体系和练习体系。

(4) **教材风格鲜明。**本套教材按 4 个专业方向统一规划, 分批组织, 陆续出版。教材的编写体现了现代教育理念, 探讨先进的教学方法。

(5) **开展教材立体化建设。**根据需要配合主教材的建设适时开发实验教材、教师参考书、学生参考书、电子参考资料等教辅资源, 为教学实现多方位服务。

我们衷心希望本系列教材能够为我国高等院校计算机科学与技术等专业的教学作出贡献, 欢迎广大读者广为选用。

“面向计算机科学与技术专业规范系列教材”编委会

面向计算机科学与技术专业规范系列教材

编委会

主任委员： 蒋宗礼

副主任委员： 王志英 钱乐秋

委员：（以姓氏拼音为序）

陈道蓄	陈 明
傅育熙	何炎祥
黄刘生	贾云得
姜守旭	李仁发
李晓明	刘 辰
马殿富	齐 勇
孙吉贵	孙茂松
吴功宜	吴 跃
谢长生	于 戈
张 钢	周兴社

秘书组： 温莉芳 刘立卿 姚 蕾

本书责任编辑： 黄刘生



前 言

什么是数据结构？数据结构又有什么作用？什么是算法？算法又有哪些行之有效的基本策略？什么是程序设计的基础？程序设计的能力又可以从哪些方面去提高？这是计算机专业的学生在学习中经常感到困惑的问题，也是利用计算机解决具体应用问题的程序员需要明白的问题，还是计算机教育工作者应该深思的问题。弄清楚了这些基本问题，以及它们之间的相互关系，无论是对学生的学习、教师的施教还是程序员的应用，都具有重要的意义和极大的帮助。

早在 1976 年，著名的计算机科学家、Pascal 程序设计语言之父、结构化程序设计首创者、1984 年图灵奖获得者沃斯(Niklaus Wirth)就提出了 $\text{Algorithms} + \text{Data Structures} = \text{Programs}$ (算法+数据结构=程序)，在这个著名的公式中，“+”生动地表达了算法和数据结构的相互作用，是程序设计的精髓；“=”言简意赅地刻画了算法和数据结构是构成计算机程序的两个关键要素。

我们认为，程序设计包括算法、数据结构和程序设计语言三部分。算法是求解问题的过程描述；数据结构是对所要求解问题中的数据的存储和算法策略实现的支持；程序设计语言是使用计算机求解问题，是在计算机上实现算法和数据结构。

本书是根据《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》的指导思想编写而成的，涵盖了“数据结构”公共核心课程的知识单元。本书共七章，分为数据结构和算法两大部分。其中，数据结构部分(第 1~4 章)按数据元素之间存在的对应关系进行划分，分为表示一对一关系的线性表、表示一对多关系的树以及表示多对多关系的图和广义表，可以以线性表作为基本数据结构，从多个方向和层面上进行拓展，引出线性表的各类变形，将栈、队列、串视为线性表的特例，将数组、广义表、树和图视为线性表的扩展，如图 1 所示；算法部分(第 5~7 章)以查找和排序算法作为常用算法，所以该部分由算法设计策略、查找和排序组成。

各章具体内容如下：

第 1 章绪论，由吴跃编写。该章通过迷宫问题给出了数据结构的两大用途：存放要处理的数据和实现算法策略。该章还定义了数据结构的概念和相关术语，给出了算法和算法分析的概念。

第 2 章线性表，由尚明生编写。线性表表示数据元素之间一对一的关系。本章以线性表作为基本数据结构，从多个方向和层面进行拓展：引出线性表的各类变形，将栈、队列、串视为线性表的特例，将数组视为线性表在维数上的扩展。

第 3 章树，由陈端兵编写。树表示数据元素之间一对多的关系。本章以二叉树的性质、存储结构和遍历算法为基础，逐步拓展到二叉树的各类变形、树和森林以及树的变形。最后举例说明了树的应用。

第 4 章图和广义表，由李树全编写。图和广义表表示数据元素之间多对多的关系。本章分别介绍了图和广义表的概念、存储结构和遍历算法，并给出了图的经典应用。

第 5 章算法设计策略，由李树全编写。本章分别介绍了直接法、分治法、贪心法、动态

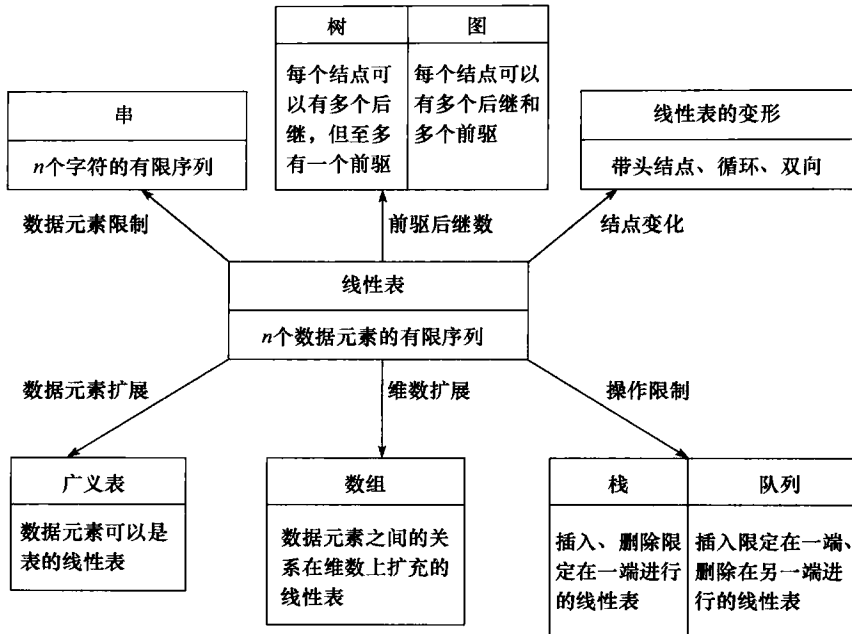


图1 数据结构关联图

规划法、回溯法、分支限界法 6 大类算法的设计策略、思想、应用特点和步骤，并举例说明了这些算法策略的应用。

第 6 章查找，由尚明生编写。在许多应用中都需要查找，可以将其视为一类基本而常用的算法，本章分别介绍了顺序表、索引表和散列表上的查找算法。

第 7 章排序，由陈端兵编写。和查找一样，在许多应用中也都需要排序，可以将其视为一类基本而常用的算法，本章分别介绍了插入、交换、选择、归并和基数 5 大类排序算法。

掌握了基本数据结构和算法策略，有助于选择和设计适合应用的数据结构，有助于设计更高效的算法，有助于提高编程能力。本书每章都有学习目标，还配有一定量的填空题、选择题、简答题和算法设计题，供读者练习。本书适用于高等院校计算机专业、信息技术类专业和软件工程相关专业的学生学习，也可供程序员参考。

最后，让我们一起一页一页地翻开这本丰富多彩的书，像一只漫游在数据结构这棵大树上的小虫，去领略各类数据结构的功用和风采，去品味各种算法策略的精妙和乐趣，去夯实计算机专业的基础，去提高编程的能力，去收获学习的果实。

编者

2009 年 11 月



教学建议

第1章 绪论(4学时)

本章结合迷宫问题介绍利用计算机求解实际问题的过程,引入数据结构的基本概念和基本术语,讲解数据结构的发展、数据结构和算法的关系以及算法的描述方法,讲解算法复杂度的基本度量标准。

通过本章的学习,让学生对数据结构有一个感性的认识,了解数据结构在解决实际问题中的应用,掌握数据结构的基本概念与表示,掌握算法的描述方法和算法时间复杂度的计算方法。

第2章 线性表(18学时)

本章开始介绍线性表的逻辑结构、基本操作方法和基本算法,讲解线性表的顺序存储结构及其基本操作方法和算法实现,讲解线性表的链式存储结构及其基本操作方法和算法实现,最后介绍线性表的几种典型应用。2.1节的重点是让学生在了解线性表的基本表示和操作的基础上,掌握线性表的应用编程。

栈和队列是线性表的扩展,在实际程序设计中有着重要的应用,2.2节和2.3节介绍栈的逻辑结构、表示和实现以及队列的逻辑结构及其链式和顺序存储结构,重点讲解栈和队列的几种实际应用编程,让学生了解数据结构在解决实际问题中的应用。

数组也可以看作线性表的扩展。2.4节的重点是让学生了解数组的顺序存储结构,稀疏矩阵的压缩存储、三元组表表示及稀疏矩阵转置、加法及乘法的实现,数组的应用——迷宫问题的编程实现。

第3章 树(8学时)

本章介绍二叉树的基本概念及基本性质、二叉树的存储结构和二叉树的基本操作(包括二叉树的遍历及其应用、二叉树的线索化),应重点掌握二叉树的三种遍历方式及基本应用。在此基础上,了解二叉树的变形(包括二叉排序树以及二叉排序树的查找、插入、删除等操作,二叉平衡树及其基本调整方法),重点掌握二叉排序树的查找、插入和删除操作;了解树和森林的基本概念和存储结构、树和森林的基本操作;了解树的变形(主要包括四叉树、八叉树、B树以及2-3树),以及在树上的查找、插入、删除操作;了解树的基本应用(包括赫夫曼编码、算术表达式的求值、堆排序以及树在决策分析中的应用),重点掌握赫夫曼树及赫夫曼编码、堆排序的过程及算法实现。

第4章 图和广义表(8学时)

本章介绍图的基本概念,让学生了解图在现实世界中的一些应用。应掌握图的常用存储结构,包括邻接矩阵、邻接表、邻接多重表、十字链表这四种存储结构;掌握图的基本遍历算法;掌握图的生成树算法、拓扑排序、关键路径和最短路径算法;掌握广义表的基本概念与存储表示,了解广义表的基本运算。通过本章的学习,进一步提高数据结构的应用能力、算法设计能力和编程能力。

第5章 算法设计策略(8学时)

数据结构与算法设计技术是密切相关的,为设计好的算法,选择适当的数据结构是非常

重要的。应了解算法分析的基本方法；掌握穷举法、递推法和迭代法这三种基本算法；重点掌握分治法、贪心法、回溯法这三种基本算法；对动态规划法、分支限界法作一般性的了解。通过本章的学习，增进程序设计的技巧，进一步提高分析问题和解决问题的能力。

第6章 查找(4学时)

本章介绍顺序表的基本概念和查找算法。应掌握顺序查找和二分查找，了解查找算法的性能分析方法；对索引表的顺序查找、二分查找作一般性的了解；掌握散列表的基本概念，掌握不同的散列函数构造方法和冲突处理方法。

第7章 排序(6学时)

在本章中，应掌握排序的基本概念和定义；掌握插入排序、交换排序、选择排序和归并排序算法及复杂度分析；了解基数排序算法及复杂度分析；掌握各种排序算法的性能比较，包括时间复杂度、空间复杂度和稳定性。

本书的总学时数为56学时，线性表、树和算法设计策略这三章内容相对较多，教师可以根据实际学时数对讲解内容作适当的调整。

本书各章最后都附有习题，教师可以根据情况，给学生留一些基本的和中等难度的习题作为课外作业。如有时间，也可以安排1至2次习题课。在习题课上可以由教师讲解以前课外作业中普遍存在的问题，也可以安排稍难一些的习题让学生在课上做出解答，然后由教师指导进行讨论，最后得出正确且较好的答案。

数据结构与算法课程课内上机为8学时，可考虑将上机实验的内容放在线性表、树和图的应用这三个部分，其他上机内容可以让学生在课外完成。

目 录

出版者的话	
序 言	
前 言	
教学建议	
第 1 章 绪论	1
1.1 计算机问题求解过程	1
1.2 迷宫问题	2
1.3 数据结构	4
1.3.1 数据结构研究的内容	4
1.3.2 数据结构概念	6
1.4 算法	10
1.4.1 算法概念及特性	10
1.4.2 算法描述	11
1.4.3 算法分析	12
1.5 本章小结	14
1.6 习题	14
第 2 章 线性表	16
2.1 线性表	16
2.1.1 线性表的定义	16
2.1.2 线性表的顺序存储	20
2.1.3 线性表的链式存储	28
2.1.4 链表的各种变形	37
2.1.5 线性表的应用	40
2.2 栈	45
2.2.1 栈的定义	45
2.2.2 栈的顺序存储	46
2.2.3 栈的链式存储	49
2.2.4 栈的应用	51
2.3 队列	55
2.3.1 队列的定义	55
2.3.2 队列的顺序存储	56
2.3.3 队列的链式存储	59
2.3.4 优先队列	61
2.3.5 队列的应用	62
2.4 数组	65
2.4.1 数组的定义	66
2.4.2 数组的表示和实现	66
2.4.3 数组的应用	77
2.5 本章小结	79
2.6 习题	80
第 3 章 树	82
3.1 二叉树	83
3.1.1 二叉树的基本概念和性质	83
3.1.2 二叉树的存储结构	86
3.1.3 二叉树的遍历	88
3.2 二叉树的变形	93
3.2.1 线索二叉树	93
3.2.2 二叉排序树	96
3.2.3 平衡二叉树	100
3.2.4 赫夫曼树及赫夫曼编码	105
3.3 树和森林	109
3.3.1 树和森林的定义	109
3.3.2 树和森林的存储结构	110
3.3.3 树和森林的基本操作	112
3.4 树的变形	115
3.4.1 四叉树	115
3.4.2 B 树	122
3.4.3 2-3 树	127
3.5 树的应用	128
3.5.1 算术表达式	128
3.5.2 堆排序	129
3.5.3 决策分析	131
3.6 本章小结	134
3.7 习题	134
第 4 章 图和广义表	137
4.1 图简介	137
4.1.1 基本概念和术语	138
4.1.2 图的应用	139
4.2 图的存储结构	140

4.2.1 图的顺序存储结构	140	5.8 本章小结	206
4.2.2 图的链式存储结构	141	5.9 习题	206
4.3 图的遍历	145	第6章 查找	209
4.3.1 深度优先遍历	145	6.1 顺序表的查找	209
4.3.2 广度优先遍历	147	6.1.1 顺序查找	209
4.4 图的应用	148	6.1.2 二分查找	211
4.4.1 最小生成树	148	6.2 索引表的查找	212
4.4.2 拓扑排序	155	6.2.1 索引表的基本概念	212
4.4.3 关键路径	158	6.2.2 索引表的顺序查找	213
4.4.4 最短路径	163	6.2.3 索引表的二分查找	215
4.5 广义表	167	6.2.4 索引表的树组织查找	216
4.5.1 广义表的定义	167	6.3 散列表的查找	216
4.5.2 广义表的存储结构	169	6.3.1 基本概念	216
4.5.3 广义表的遍历	170	6.3.2 散列函数	217
4.5.4 广义表的运算	172	6.3.3 冲突处理	218
4.6 本章小结	175	6.3.4 散列查找与性能分析	220
4.7 习题	175	6.4 本章小结	223
第5章 算法设计策略	179	6.5 习题	223
5.1 算法分析技术	179	第7章 排序	226
5.2 直接法	183	7.1 排序的基本概念	226
5.2.1 穷举法	183	7.2 插入排序	227
5.2.2 递推法	184	7.2.1 直接插入排序	227
5.2.3 迭代法	185	7.2.2 二分插入排序	228
5.3 分治法	186	7.2.3 希尔排序	229
5.3.1 分治法的基本思想	187	7.3 交换排序	230
5.3.2 斯特拉森矩阵乘法	188	7.3.1 冒泡排序	230
5.4 贪心法	190	7.3.2 快速排序	232
5.4.1 贪心法的基本思想	190	7.4 选择排序	234
5.4.2 背包问题	191	7.4.1 简单选择排序	234
5.5 动态规划法	193	7.4.2 树形选择排序	235
5.5.1 动态规划法的基本思想	193	7.5 2路归并排序	236
5.5.2 矩阵连乘问题	196	7.6 基数排序	237
5.6 回溯法	198	7.6.1 多关键字排序	237
5.6.1 回溯法的基本思想	198	7.6.2 链式基数排序	238
5.6.2 回溯法的形式化描述	199	7.7 各排序方法的比较	241
5.6.3 八皇后问题	200	7.8 本章小结	241
5.7 分支限界法	202	7.9 习题	242
5.7.1 分支限界法的基本思想	202	参考文献	245
5.7.2 货郎担问题	203		



绪 论

“数据结构”是计算机科学技术专业的一门基础理论课程，是计算机学科的核心课程之一。在计算机科学技术各个领域，选择合适的数据结构都是一个重要问题。通过数据结构与算法的学习，读者能进一步提高软件设计与程序编写的能力，提高应用计算机技术解决实际问题的能力。

著名的瑞士计算机科学家、Pascal 程序设计语言之父、结构化程序设计首创者、1984 年图灵奖获得者沃斯 (Niklaus Wirth) 于 1976 年提出了 $\text{Algorithms} + \text{Data Structures} = \text{Programs}$ (算法+数据结构=程序)，在这个著名的公式中，“+”生动地表达了算法和数据结构的相互作用，是程序设计的精髓；“=”言简意赅地刻画了算法和数据结构是构成计算机程序的两个关键要素。计算机程序是使用计算机求解问题，问题的有效求解涉及算法和数据结构这两个不可分割的部分。算法是求解问题的过程描述，数据结构是对所要求解问题中的数据的存储和算法策略实现的支持。什么是数据结构？什么是算法？有哪些常用的数据结构？又有哪些经典的算法策略？这就是本书的主要内容。让我们一起一页一页地翻开这本丰富多彩的书，像一只漫游在数据结构这棵大树上的小虫，去领略各类数据结构的功用和风采，去品味各种算法策略的精妙和乐趣。

本章将概括地介绍数据结构与算法设计的基础知识，主要包括：

- 结合迷宫问题介绍利用计算机求解实际问题的过程。
- 介绍数据结构的概念和基本应用。
- 介绍算法的基本概念和对算法的分析方法。

通过这一章的学习，我们将掌握数据结构与算法的概念，了解数据结构的基本应用，掌握评价算法的基本方法。

学习目标：

- 了解迷宫问题的求解过程，掌握计算机求解问题的基本步骤
- 了解数据结构在实际问题中的应用，掌握数据结构的基本概念与表示
- 了解算法描述的方法，掌握算法的基本概念与评价方法

1.1 计算机问题求解过程

计算机问题求解过程通常分为五步：第一步是对要求解问题的理解，了解问题的输入、要求和输出；第二步是数据结构设计，要选择或设计能有效表示和存储应用问题中所涉及的数据对象的数据结构，同时还要选择或设计能支持算法策略实现的数据结构；第三步是算法

设计，包括选择算法策略、用适当的方式描述和逐步细化算法步骤；第四步是对所设计的算法进行分析，发现有改进完善之处，返回第二步，重新选择或设计数据结构，重新设计算法；第五步是算法的程序实现，用某种计算机程序设计语言，定义数据结构，编写实现算法的代码，在计算机上调试和运行程序。这个过程如图 1-1 所示。

这里需要强调以下三点：

1) 第二步中选择或设计数据结构以及第三步中选择算法策略都是非常重要的，是影响算法性能的关键，也是本书的重点。

2) 算法与程序是有区别的。算法是求解问题的一种方法或一个过程，是将输入按要求转换成输出的一个变换。通常，对一个待求解的问题有多种实现算法。程序是用某种程序设计语言对算法的具体实现。程序和算法之间的主要区别在于有穷性和描述方法：

- 程序可以是无穷的，算法是有穷的。例如，操作系统是系统软件，是一组程序而不是算法，因为开机后操作系统就会不停地运行，哪怕没有作业需要处理，它也在执行，以等待进程。
- 程序是用程序设计语言描述的，在计算机上可以执行；算法除了可以用程序设计语言描述以外，还可以用框图、自然语言等方式描述。

3) 算法与数据结构是有联系的：

- 一方面，算法所求解问题的对象需要用适当的数据结构存储到计算机中，算法才能对其进行操作和处理；另一方面，算法本身需要适当的数据结构来支持算法策略的实现。
- 一个高效的算法常常是因为选用或设计了一个好的数据结构，算法与数据结构是计算机问题求解的关键技术。

下面我们以迷宫问题为例，来初步认识什么是数据结构和算法。

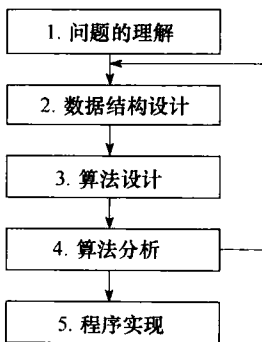


图 1-1 计算机求解问题的过程

1.2 迷宫问题

1. 迷宫问题描述

在资源勘探、战争或游戏中，都会有类似的情景，即在给定的空间（如森林、山洞、沙漠等）中搜索特定目标（如出路、人或物等）。这类问题都可以归结为迷宫问题，即从出发点（入口）开始，在给定的空间中，沿可行的路径进行探索，直到达到目标（出口）。

如何利用计算机来求解迷宫问题？这需要解决以下几个问题：

- 如何表示给定的空间和可行的路径？
- 如何表示入口和出口？
- 当有多条可行的路径时，如何选择？
- 当某条路径在某一点再没有可行之路时，如何处理？

前两个问题属于数据结构选择和设计，后两个问题涉及算法设计。

2. 迷宫问题表示

我们可以用一个 m 行 n 列的二维数组 $maze[m][n]$ 来表示迷宫空间（或称迷宫地图），并约定：

- 当数组元素 $maze[i][j]=0$ 时，表示通路；当数组元素 $maze[i][j]=1$ 时，表示不通。
- 入口为左上角 $maze[1][1]$ ，出口为右下角 $maze[m][n]$ 。

在这样表示的迷宫中，有以下三类点：

- 角点，有 4 个： $maze[1][1]$ 、 $maze[1][n]$ 、 $maze[m][1]$ 和 $maze[m][n]$ ，它们都有 3 个探索方向。例如， $maze[1][1]$ 点只可以沿正东方向到 $maze[1][2]$ 点、沿东南方向到 $maze[2][2]$ 点、沿正南方向到 $maze[2][1]$ 点。
- 边点，有 $(m-2+n-2)*2$ 个，它们都有 5 个探索方向。例如， $maze[2][1]$ 点可以沿正东方向到 $maze[2][2]$ 点、沿东南方向到 $maze[3][2]$ 点、沿正南方向到 $maze[3][1]$ 点、沿东北方向到 $maze[1][2]$ 点、沿正北方向到 $maze[1][1]$ 点。
- 中间点，有 $(m-2)*(n-2)$ 个，它们都有 8 个探索方向。

为了使问题简单，我们可以在原来的迷宫地图四周都扩展一个点，即增加两行和两列，用 $maze[m+2][n+2]$ 来表示迷宫地图，并将迷宫四周增加点的值全部置为 1，表示是墙，不能通行。这样做使得原迷宫地图中的所有点都成为了中间点，不用再判断当前点是角点、边点还是中间点，每个点的探索方向均为 8 个，如图 1-2 所示。

入口 (1, 1)	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1
2	1	1	0	1	0	1	1	1	1	1
3	1	0	1	0	0	0	0	0	1	1
4	1	0	1	1	1	0	1	1	1	1
5	1	1	0	0	1	1	0	0	0	1
6	1	0	1	1	0	0	1	1	0	1
7	1	1	1	1	1	1	1	1	1	1
									出口 (6, 8)	

图 1-2 迷宫地图

迷宫问题可定义为：

```
#define m 6 /* 迷宫的实际行 */
#define n 8 /* 迷宫的实际列 */
int maze[m+2][n+2]; /* 四周扩展后的迷宫地图 */
```

3. 迷宫问题求解

对于某一点 (x, y) ，它有 8 个可以探索的方向，如果沿正东方向可行的话，即 $maze[x+1][y]=0$ ，则可以到达 $(x+1, y)$ 点；如果沿东南方向可行的话，则可以到达 $(x+1, y+1)$ 点；……这 8 个探索方向的坐标变化如图 1-3 所示。

$x-1, y-1$	$x, y-1$	$x+1, y-1$
$x-1, y$	x, y	$x+1, y$
$x-1, y+1$	$x, y+1$	$x+1, y+1$

图 1-3 8 个探索方向的坐标变化

我们可以定义一个探索方向的增量数组，用来记载在 (x, y) 点按某个方向探索时坐标 x 和 y 的变化量。

定义 $DeltaXY$ 数组为：

```
typedef struct
```

```

{
    int x,y
} item;
item DeltaXY[8];

```

若规定探索方向为从正东方向开始沿顺时针方向依次进行探索，则 ΔXY 数组内容如图 1-4 所示。

$\Delta XY[0]$ 表示正东方向 x 和 y 的增量， $\Delta XY[1]$ 表示东南方向 x 和 y 的增量， $\Delta XY[2]$ 表示正南方向 x 和 y 的增量，……， $\Delta XY[7]$ 表示东北方向 x 和 y 的增量。

至此，我们用二维数组 $maze[m+2][n+2]$ 来表示迷宫解决了迷宫地图的存储；还用一维数组 $\Delta XY[8]$ 来记载 8 个探索方向的坐标增量，将 8 个探索方向数字化为 0~7，并将向下一点前进的操作统一为“当前点的坐标+沿该探索方向的增量”，这样就得到下一点的坐标。其中， $maze[m+2][n+2]$ 和 $\Delta XY[8]$ 都是数据结构的选择或设计。

x	y
1	0
1	1
0	1
-1	1
-1	0
-1	-1
0	-1
1	-1

图 1-4 探索方向的增量数组

当到达了某点并沿该点的 8 个方向都探索后仍无路可通行时，需要从该点返回到前一点，再从前一点选择下一个方向继续进行探索，即需要回退到前一点和前一点当前探索的方向。因此，我们需要保留依次到达的各点的坐标和到达该点的方向。

我们可以选用一个数据结构——栈，栈中的元素是由行号 x 、列号 y 和到达该点的方向 d 组成的三元组 (x, y, d) ，其中 x 为到达点的横坐标或行号， y 为到达点的纵坐标或列号， d 为 0~7 中的一个数字，表示到达坐标为 (x, y) 的点的方向。例如，从 $(2, 2)$ 点沿东南方向到达 $(3, 3)$ 点时，在栈中要记录一个三元组 $(3, 3, 1)$ 。

另外，还需要防止重复到达某点，以避免重复探索，在迷宫中兜死圈子，为此需要记载已到过的点。可以用以下两种方法解决该问题：

- 一种方法是设置一个标志数组 $mark[m][n]$ ，它的所有元素都初始化为 0，一旦到达某点 (i, j) 之后，将标志数组 $mark$ 的对应下标（即 $mark[i][j]$ ）置为 1，在探索中，当所探索的点 (i, j) 相对应的 $mark[i][j]=0$ 时，才进入该点，当所探索的点 (i, j) 相对应的 $mark[i][j]=1$ 时，表明已探索过，不需要再进入该点了。
- 另一种方法是当到达某点 (i, j) 后，在迷宫地图的该点坐标上标记特殊值，例如，将 $maze[i][j]$ 置 -1，以区别未到达过的点。

在迷宫问题求解的例子中，我们可以看到数据结构有以下两大用途：

- 一是用于存放要处理的数据，如迷宫地图。
- 二是用于实现算法策略，如迷宫例子中探索方向的增量数组、回溯的栈、避免重复探索的标志数组或特殊标记。

1.3 数据结构

1.3.1 数据结构研究的内容

数据结构课程是计算机专业的核心课程之一，也是计算机专业知识的基石之一。下面先看几个简单的例子，初步了解数据结构这门课程涉及的主要内容。

【例 1.1】 字符串含义理解

对字符串“02232009”，若不知道数字之间的结构或联系，我们将很难准确地理解它所表达的意思。但如果将其分组为“02/23/2009”，则可将其看作一个日期，即2009年2月23日。同样，考虑在一个字节的存储单元中的二进制串“01100001”，如果看作一个整体，可以看作整数97，也可以看作字符“a”的ASCII码。

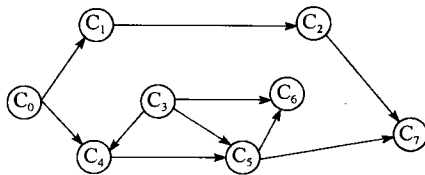
这个例子表明：杂乱的数据或没有结构的数据不能有效地表达和交流信息。

【例 1.2】 教学计划编排

一个教学计划包含许多课程，有些课程之间存在一定的先后关系，在编排教学计划时，必须按这些课程的先后顺序进行。用圆圈表示课程，圆圈之间的有向边表示课程的优先关系。如果课程*i*是课程*j*的先修课，则存在一条从圆圈*i*指向圆圈*j*的有向边。这样，可以用一个图形来表示课程之间的优先关系，如图1-5所示。

课程代号	课程	先修课程
C ₀	高等数学	
C ₁	数字逻辑	C ₀
C ₂	计算机组成原理	C ₁
C ₃	高级语言程序设计	
C ₄	离散数学	C ₀ , C ₃
C ₅	数据结构	C ₃ , C ₄
C ₆	编译技术	C ₃ , C ₅
C ₇	操作系统	C ₂ , C ₅

a) 计算机专业基本课程表及优先关系



b) 表示优先关系的图形

图 1-5 课程优先关系

这个例子表明：数据元素（课程）之间存在一定的关系，该例中是课程之间的修课先后顺序，即先修课与后继课的顺序关系。数据结构就是要研究数据元素之间的各种关系。

【例 1.3】 电话簿查询

通常每个人的电话号码簿中至少存在一条记录，每个记录至少包括姓名和电话号码两部分。这些记录在电话号码簿中有多种存放方式，如可以随机填写，也可以按照姓名的字典顺序进行排列。如果要求在电话号码簿中查找某人的电话号码，则根据记录的不同组织方式，可能采用不同的查找方法。如果电话号码是随机填写的，则只能采取顺序查找、逐一比较的方法；如果电话号码是按照姓名的字典顺序排列的，则既可以采取顺序查找方法，也可以使用先找到姓然后在该姓中找名的字典式快速查找方法。

这个例子表明：数据的不同组织方式将直接决定其操作的方法。由于实现目标的方法可以很多，因而需要对各种方法进行比较和评价。数据结构就是要研究可能存在的组织方式，以及不同的组织方式对算法效率的影响。

【例 1.4】 图书目录管理

通常书目包含的主要内容有书名、作者、书号、类别、出版日期等，对书目可以进行一些操作，例如，新书入库时，要登记建立书目，并将新建的书目归类插入到合适的位置；使用书目时，通过对书目的查找，可以得知该书在书库中的存放位置。

操作和数据的组织密切相关。例如，对于查找操作，如果书目的组织没有规律，则只能逐一进行比较；如果按照某种顺序组织，比如按照书名的字典顺序或分类号，则查找将快得多。