



面向21世纪课程教材  
Textbook Series for 21st Century

# 算法与数据结构

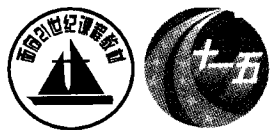
## —C语言描述

### (第3版)

张乃孝 陈光 孙猛 编著



高等教育出版社  
HIGHER EDUCATION PRESS



面向 21 世纪课程教材  
普通高等教育“十一五”国家级规划教材  
普通高等教育精品教材

# 算法与数据结构

——C 语言描述

Suanfa yu Shuju Jiegou——

C Yuyan Miaoshu

(第 3 版)

张乃孝 陈 光 孙 猛 编著



高等教育出版社·北京  
HIGHER EDUCATION PRESS BEIJING

## 内容提要

本书以数据结构为主线、算法为辅线组织教学内容。全书共10章,内容包括绪论、线性表、字符串、栈与队列、二叉树与树、集合与字典、高级字典结构、排序、图、算法分析与设计。本书第1版为“面向21世纪课程教材”,2004年被评为“北京市高等教育精品教材”,第2版为普通高等教育“十一五”国家级规划教材,2007年获“普通高等教育精品教材”奖。

本书体系完整、概念清楚、内容充实、取材适当,采用“数据结构作为抽象数据类型的物理实现”观点,既提高了抽象数据类型在本课程教学中的地位 and 作用,又突出了自身的教学重点。本书在讲解知识的同时,重视能力的培养,以提高学生运用知识解决实际问题的能力。新版对第2版教材中许多算法进行了改进,力求为读者提供一套具有良好C语言风格、更便于教学的程序代码,以期帮助学生从中体会到算法的魅力和C语言编程的艺术,提高学生的学习兴趣。同时,新版内容也适当地提高了知识的深度和广度,完全覆盖了最新考研大纲的内容要求。

本书许多知识模块具有一定的独立性和相关性,因此不同专业和不同水平的读者可以根据需要组合使用。本书既可以作为计算机专业本科“数据结构”课程教材,也可以作为理工科有关专业本科和计算机专业专科相关课程的教材或考研参考书。

## 图书在版编目(CIP)数据

算法与数据结构: C语言描述/张乃孝, 陈光, 孙猛编著. -- 3版. -- 北京: 高等教育出版社, 2011.6  
ISBN 978-7-04-034136-2

I. ①算… II. ①张… ②陈… ③孙… III. ①算法分析-高等学校-教材 ②数据结构-高等学校-教材 ③C语言-程序设计-高等学校-教材 IV. ①TP301.6 ②TP311.12 ③TP312

中国版本图书馆CIP数据核字(2012)第008552号

策划编辑 倪文慧 责任编辑 倪文慧 封面设计 于文燕 版式设计 王艳红  
责任校对 殷然 责任印制 刘思涵

出版发行 高等教育出版社  
社址 北京市西城区德外大街4号  
邮政编码 100120  
印刷 肥城新华印刷有限公司  
开本 787mm × 1092mm 1/16  
印张 24  
字数 440千字  
购书热线 010-58581118  
咨询电话 400-810-0598

网 址 <http://www.hep.edu.cn>  
<http://www.hep.com.cn>  
网上订购 <http://www.landaco.com>  
<http://www.landaco.com.cn>  
版 次 2002年9月第1版  
2011年6月第3版  
印 次 2011年6月第1次印刷  
定 价 39.00元(含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。  
版权所有 侵权必究  
物料号 34136-00

---

# 数据结构三十年(自序)

“数据结构”的概念出现在 20 世纪 60 年代,在西方发达国家正式设为本科课程则是 20 世纪 70 年代的事情。1978 年,北京大学计算机系成立。本人任数据库教研组长期间,到北京国家图书馆查阅了有关外文资料,提出了第一个《数据结构》教学大纲,并且和杨冬青等老师一起编写了第一本《数据结构》讲义。1980 年起,这本讲义在校内外开始试用,受到广泛好评,又经多次修改最终于 1987 年由高等教育出版社出版<sup>[1]</sup>。

这本讲义主要参考了当时有影响的几本著作<sup>[4,5,6,7,8]</sup>。在数据结构的许多概念还十分模糊的阶段,对讲义中的基本概念反复查证和推敲,给出尽可能准确一致的解释。特别是对“数据结构”这个最基本的概念,国外不同教材的说法千差万别,比较严格的有 H. Maurer<sup>[7]</sup>和 C. C. Gotlieb 的两种形式化<sup>[8]</sup>定义。经过认真比较,研究各种观点的含义,我们在讲义中把“数据结构”定义为:“按某种逻辑关系组织起来的一批数据,按一定的存储表示方式把它存储在计算机中,并在这些数据上定义了一组运算”。全书按“逻辑结构、存储结构、运算和应用”4 个层次的树型结构组建架构<sup>[16]</sup>,概念清楚、内容丰富、体系完整。由于当时没有适合的高级语言,所以自己定义了一种简单明了的伪语言,用它描述算法。

1985 年,“北京市自学考试委员会”开设计算机专业。作为数据结构课程的考试委员,我邀请杨冬青和邵维忠两位老师共同编写了自学考试教材《数据结构基础》<sup>[19]</sup>。这本书以 1987 版《数据结构》教材内容为基础,参考了 A. V. Aho 等教授的教材<sup>[11]</sup>,改用以数据结构为主线、算法为辅线结构编写,使得全书内容更加紧凑、重点更加突出,并选用了当时十分流行的 Pascal 语言描述算法。在教材开始,用一个实际问题生动地介绍了使用计算机求解的全过程,第一次在国内《数据结构》教材中,引入了“抽象数据类型”的概念,以帮助读者理解数据结构和算法的地位和作用。为了更加便于教学,在教材中增加了从算法思想到代码实现逐步求精的具体例子,以帮助读者解决“编程难”的问题。

20 世纪 90 年代,面向对象的语言开始流行。我们查阅了国外众多 C++ 描述数据结构的教材,发现其中不少名义上使用是 C++ 语言,实际上并没有采用面向对象的方法,也没有使用到 C++ 中面向对象的高级机制,因此,我与裘宗燕老师合作编写了《数据结构——C++ 与面向对象的途径》<sup>[2]</sup>。这本教材参考了 T. A. Budd 的工作<sup>[12]</sup>和自考教材《数据结构基础》一书的结构,在讲解数据结构的同时,增加了面向对象知识的介绍和 C++ 的使用方

法,根据面向对象的方法,由简到繁、从易到难,引入各种抽象数据类型的概念和实现,并用类图描述了各种结构之间的内在联系。该教材经过5年的教学实践,取得了许多成功的经验。同时在使用过程中我们也发现了两个问题:一个问题是由于该教材内容多、难度大,对于基础好的学生,只要开始能顺利接受面向对象的思想,学会C++语言编程,以后就会越学越好,越学越有兴趣;否则就可能越学越困难,所以最后学生两极分化明显;另一个问题是由于使用了良好的面向对象的C++描述,程序表面的可读性很好,但内涵十分丰富,例如各种构造函数和析构函数的自动选择和运行,各种继承和多态功能的动态处理等,所以要具体分析其时间和空间的代价往往比较困难。

数据结构教材必须反映计算机科学的最新发展,这是保持教材先进性的关键。1998年,北京大学聘请我主持全校理科主干基础课“算法与数据结构”时,我们重新编写了一本适合理科各专业的教材《算法与数据结构》,即本书的第一版。该教材充分吸取了前三本教材的经验,在内容、结构、方法和描述等方面既有继承、融合又有创新。

如何与时俱进地理解和定位“数据结构”,是编写本教材时面对的最具挑战性的问题。经过认真调查研究,参考了Kurt Mehlhorn等人的著作<sup>[23]</sup>,我们提出把“数据结构”理解为“抽象数据类型的物理实现”。这一新观点突出了抽象数据类型在数据结构教学中的地位,使读者可以从更高的层次理解数据结构与算法的关系;同时也包含了数据结构与面向对象技术的内在联系。基于上述观点,该教材的主要内容是在介绍各种典型的抽象数据类型的基础上,重点讨论它们各种不同的存储表示方式和操作的实现算法。提出“物理实现”的意图是强调本课程关心的“实现”应具体到可以用计算机的两个最重要的物理量(主机的运行时间和内存的存储空间)来权衡。

在教学方法上,该教材更加注意将讲解知识和培养能力并行:不但讲解一个抽象数据类型可以采取的常用表示形式,同时还指出不同表示的特点和各种表示使用的环境;不但讲解如何在选定的存储表示上,正确实现抽象数据类型要求的各种操作,同时还强调不同存储表示对算法效率的影响;不但讲解单个算法的设计和实现方法,同时也强调同类算法之间的共性,并且在教材最后对学到的主要算法进行了总结,以提高学生使用学到的知识去解决实际问题的能力。

在算法的描述方式上,该教材选择了C语言。因为C语言虽然是一个小语言,但具有丰富的表达能力,这使它简单、易学,又能满足基本的教学需求。另外,C语言是一个过程语言,用它描述的算法语义清晰、确定可行。特别重要的是,它比较低级,使得用C描述的算法,其时间和空间代价分析最直观、准确。算法程序是没有“标准答案”的,为了给读者提供一套具有C语言风格并更易于理解的程序代码,该教材在第3版中又对算法进行了许多改进,力求使读者从中体会到算法的魅力和C语言编程的艺术。

在教材设计方面,第3版教材更加突出各知识模块的独立性和相关性,来自不同学科专业的学生可根据不同教学目标的实际需要组合选用。另外,为满足考研读者的需要,第3版

教材又根据教育部制订的考研大纲要求,适当提高了知识的深度和广度。

《算法与数据结构》教材初稿由陈光、刘筠、韩玉真等老师与我共同编写,2002年正式出版,2006年修订,第3版教材的修订工作由陈光和孙猛两位老师和我一起共同完成。两年前,本人年近古稀,告别杏坛,在最后一次为北京大学讲授“数据结构”课程时,学校对讲授过程进行了全程录像,现把它和修改后的课件一起随第3版教材发行,希望对读者有所帮助。

光阴似箭,日月如梭,本人围绕数据结构著书立说、科研讲学的三十多年正是数据结构在中国从引介到发展的三十多年。本人的经历和兴趣在大时代的背景下潮汐,使我的一生与数据结构结下不解之缘。在1978年前,我曾有十年“系统软件”开发经验,成功研发了多个编译系统、操作系统和诊断系统,因而一接触“数据结构”就产生了极大兴趣,也能够很快体会到其中的真谛;加上长期从事“程序设计语言”和“软件理论”的研究,教学与科研相辅相成、互相促进<sup>[15][16][17][18][21]</sup>,促使我在完成第一本数据结构教材之后,结合教学实践与时俱进、精益求精,不断在教材建设的道路上探索与前进,推出了多本有代表性的作品,也取得了一些成绩。

回顾三十多年来的经历,本人心甘情愿地为写好一本书、讲好一门课付出自己半生的心血,安心于本职工作,任何凡是教学的需要,只要我能做到,一定努力做好,无怨无悔。长期工作在教学第一线,看到青年学子极高的学习热情、强烈的求知渴望,听到他们对于本课程的充分肯定和殷切的期望,都让我深深感动,在辛勤耕耘的同时,也享受着授业解惑的快乐,体会到人生的价值。

张乃孝 于蓝旗营

2012年1月

---

## 第 2 版前言

关于算法的研究已经有数千年的历史。计算机的出现,使得用机器自动解题的梦想成为现实,人们可以将算法编写成程序交给计算机执行,使许多原来认为不可能完成的算法变得实际可行。数据结构的概念最早在 1966 年由 C. A. R. Hoare 和 N. Wirth 提出。大量关于程序设计理论的研究表明:对大型复杂程序的构造进行系统而科学的研究,必须对这些程序中所包含的数据结构进行深入的研究。事实上,程序就是在数据的某些特定的结构和表示的基础上对于算法的描述。不清楚解决问题的算法就无法决定如何组织数据,反之算法的实现又在很大程度上依赖于数据的具体组织。简言之,算法与数据结构是程序设计中相辅相成、不可分割的两个方面。因此,1976 年 N. Wirth<sup>[6]</sup>用“算法 + 数据结构 = 程序”这个公式表达了算法与数据结构的联系和它们在程序中的地位。

抽象数据类型起源于 20 世纪 70 年代,可以定义成:有一定行为(操作)的抽象(数学)类型。它抽象出数据类型的使用要求,而把它的具体表示方式和运算的实现细节都隐藏起来。它支持数据类型的实现与使用分离的原则,是一种十分有效的对问题进行抽象与分解的思维工具,后来发展成为面向对象技术与方法的主要理论基础。

从抽象数据类型的观点出发,数据结构可以理解成为“抽象数据类型的物理实现”。对于数据结构的学习和研究,主要围绕两个问题:一个是如何具体表示抽象数据类型中的数学模型;另一个是如何给出抽象数据类型中需要的操作的具体实现。用上述观点,可以从更加抽象的角度理解数据结构与算法的关系,也容易解释数据的逻辑结构、存储结构与运算三者的关系。

### 目的与动机

“数据结构”(或称“数据结构与算法”、“算法与数据结构”)是计算机科学的基础课程,它被独立列入大学本科的教学计划已有 30 多年的历史。其主要目的是使读者全面地理解数据结构和算法的概念,掌握设计数据结构与算法的主要原理和方法,比较不同数据结构和算法的特点,通过学习和实践,提高学生使用计算机解决问题的能力。

1998 年,北京大学把“算法与数据结构”作为第一批理科各系主干基础课列入学校的教学计划。经信息学院和数学学院共同推荐,著者连续六年被校长聘任为该课程的全校主持

人,负责组织理科各院系的主讲老师制定教学大纲、编写教材和教学辅导用书、交流教学经验等工作。

本书的第1版就是在这个背景下,邀请了北京大学物理学院、化学学院和计算中心的几位主讲老师与著者共同编写的。从2000年开始,本书作为北京大学主干课“算法与数据结构”的教材在校内试用,2002年得以在高等教育出版社正式出版,并得到广大校内外老师和学生的支持和鼓励,2004年被评为“北京市高等教育精品教材”。为了回报广大读者,使得这本书能够更上一个台阶,成为过得硬的“精品”教材,在高等教育出版社的大力支持下,著者决定认真修改后出此新版。

## 第2版的主要修改

- 采用“数据结构是抽象数据类型的物理实现”观点,改写了第1章和其他有关部分,提高了抽象数据类型在本书中的地位和作用;
- 增加了“优先队列”、“矩阵”、“广义表”和“动态存储管理”等内容;
- 增加了集合的“位向量”表示和字典的“字符树”表示,把原来第6章加以扩充后,分成两章;
- 增加了“外存数据的组织”、“散列文件”和“外排序”等基本文件知识的介绍;
- 增加了小标题,使得内容的重点更加突出,并且对于书中的叙述进行了加工调整,提高了全书的层次感、简明性和可读性;
- 进一步推敲了基本概念的表述,增加了主要概念的索引;
- 补充了习题;
- 删除了附录。

## 本书的组织

本书以数据结构为主线、算法为辅线组织教学内容。修改后,全书共分以下10章。

第1章绪论。首先通过一个实例讲解使用计算机处理实际问题的过程,然后对抽象数据类型、数据结构和算法等分别展开讨论。通过本章学习,可以对全书有个鸟瞰。

第2章线性表。首先介绍线性表的逻辑结构,引入线性表的抽象数据类型,接着具体给出两种线性表的实现,顺序表和链表。然后介绍一个线性表的应用实例。最后,作为线性表的扩展,讨论了矩阵和广义表的表示,并且介绍了结点的动态分配和回收等问题。

第3章字符串。介绍了字符串的抽象数据类型和表示方法,重点讲解无回溯模式匹配算法。

第4章栈与队列。除了讨论栈和队列的概念、抽象数据类型、表示方法和实现算法外,还将给出一些常见的应用例子。

第5章二叉树与树。首先引入二叉树的概念和抽象数据类型;具体给出两种二叉树的实现;介绍了二叉树的两个重要的应用实例。然后,介绍树的基本概念和抽象数据类型,讨论了树的实现问题;最后讨论了二叉树与树林的相互对应关系。

第6章集合与字典。首先介绍集合的概念、抽象数据类型、位向量表示法和实现算法;然后讨论字典的相关内容。重点介绍了字典的顺序表示和散列表示。

第7章高级字典结构。首先论述了字典与索引的关系,然后进一步讨论字典的其他实现。包括字符树、二叉排序树和多级索引结构。其中多级索引结构主要用于组织外存文件的索引。



第8章排序。介绍了10种排序方法,讨论了这些算法的复杂性及稳定性。

第9章图。介绍了图的基本概念和抽象数据类型,讨论了图的周游方法和图的相邻矩阵及邻接表的表示方法,重点讲解求图的最小生成树(林)和求图中结点间最短路径等算法,最后介绍了两个典型的应用。

第10章算法分析与设计。主要给读者概括地介绍算法的分析和设计的主要技术,以便于将本书所学到的算法归类整理,达到开阔思路、提高观点、增强兴趣的目的。

## 读者对象与使用方法

本书可以作为各种类型大学的本科和专科的“数据结构”、“数据结构与算法”或者“算法与数据结构”等课程的教材。本书在编写中注意到知识模块的独立性和相关性,不同专业的学生可以根据不同的需要进行组合使用。

### 基础部分

- 1 第1章绪论(除去1.3.4、1.4.2和1.4.3)。
- 2 第2章线性表(除去2.4、2.5和2.6)。
- 3 第4章栈与队列(除去4.3和4.6)。
- 4 第5章二叉树与树(除去5.4、5.5、5.6和5.7)。
- 5 第6章集合与字典(除去6.2.1和6.5.4)。
- 6 第8章排序(除去8.2.3、8.2.4、8.3.2、8.5和8.6.2)。
- 7 第9章图(除去9.5.2、9.6和9.7)。

### 提高部分

- 8 应用实例:分散在2.4、4.3、4.6、5.4、9.6和9.7等节中。
- 9 增强基础:学习在基础部分各章中被除去的那些(有关文件的内容除外)内容。
- 10 第3章字符串。
- 11 第7章高级字典结构。

### 任选部分

- 12 文件内容:分散在1.3.4(外存数据的组织)、6.5.4(散列文件)、7.6(索引文件)和8.6.2(外排序)。
- 13 第10章算法分析与设计。

除去基础部分中的内容应该顺序学习以外,其他部分均可以灵活安排。著者建议:

◆ 对于生物、化学和医农等专业的本科学生的相关课,可以主要学习1~5的内容和6~8中部分内容。

◆ 对于一般理工科的本科学生,可以主要学习1~9的内容和10~11中部分内容。

◆ 对于信息或计算机(或者与它们要求相似的)专业的本科学生,可以主要学习1~11的内容和12~13中部分内容。

此外,在使用本书时,请读者注意以下几点:

**语言** 考虑到目前国内“计算概论”(或“基本程序设计”)课程的教学情况,同时也因为用C语言描述算法十分有效,分析使用C语言描述的算法又比较直观,因此本书采用C语言作为描述语言,并且假设读者都具备了这方面的基础知识。

**索引** 为了便于使用,对于比较重要的概念都建立了对应的索引,读者可以利用本书最后的索引查阅。

**习题** 学习算法与数据结构必须理论与实践相结合,所以在理解书本知识的同时,应该尽可能做更多的习题。在本书的每章最后都配备了比较丰富的习题,根据不同的目的,分成复习题、算法题和应用(上机)题三种。这些题目的参考答案可以在本书参考文献[3]中找到。对于初学者而言,最难完成的是算法题。设计算法的关键在于设计的思路,不同的思路产生出不同的算法。读者通过自己的学习、思考和实践,就能从中体会到算法与数据结构的真谛,提高自己分析问题和解决问题的能力。

**学时** 根据著者的经验,对于比较熟悉 C 语言的对象,讲授全部内容需要 60~70 学时。

## 感谢

此书能够顺利与读者见面,首先应该感谢的是长期支持和鼓励著者从事教学和研究的北京大学(特别是数学学院和计算机系)。能够在北大这片沃土上,实践著者“追求真理、探索人生”的理想是著者今生的最大满足。为此,著者数十年来孜孜以求,从来不敢懈怠;平平淡淡,确也无怨无悔。借此机会略表感谢与怀念之情。

感谢在近二十多年中,与著者合作编写教材的许多老师(按照时间顺序):唐世渭、杨冬青、刘玉明、许卓群、邵维忠、孙玉方、裘宗燕、韩玉真、陈光和刘筠等。在与他们的合作过程中,著者不仅仅学到许多的知识,同时也学到了许多为人的道理。其中裘宗燕教授还阅读了本书的大部分修改稿,提出许多中肯的意见。

衷心感谢曾经和将要使用本教材的老师、同学和所有读者,你们的每条宝贵意见都是对著者最大的支持与鼓励。

十分感谢高等教育出版社计算机分社的刘建元社长和刘茜、康兆华、耿芳以及所有相关的工作人员,著者的几本主要教材,都是在他们的支持下得以顺利出版。

非常感谢著者所有的亲人,没有他们的理解、支持和关爱,著者很难能够平安度过这坎坷的前半生。特别要感谢的是赵素兰女士,在近四十年的共同生活中,是她的无私奉献构筑了如今幸福美满的家庭。最后,真诚地期望这个世界变得更加美好;衷心地祝愿我们的子孙后代生活得更加快乐。

张乃孝

znx@pku.edu.cn

2005年6月11日

# 目 录

<b>第 1 章 绪论</b> .....	(1)	2.3 链接表示 .....	(39)
1.1 从问题到程序 .....	(1)	2.3.1 单链表表示 .....	(39)
1.1.1 问题分析与抽象 .....	(2)	2.3.2 单链表上运算的实现 .....	(41)
1.1.2 程序的设计与实现 .....	(4)	2.3.3 分析与比较 .....	(44)
1.2 抽象数据类型 .....	(6)	2.3.4 单链表的改进和扩充 .....	(45)
1.2.1 什么是抽象数据类型 .....	(7)	2.4 应用举例 .....	(48)
1.2.2 意义与作用 .....	(7)	2.4.1 Josephus 问题 .....	(48)
1.2.3 举例 .....	(8)	2.4.2 采用顺序表模拟 .....	(49)
1.3 数据结构 .....	(9)	2.4.3 采用循环链表模拟 .....	(50)
1.3.1 什么是数据结构 .....	(9)	2.5 矩阵 .....	(53)
1.3.2 数据结构的分类 .....	(10)	2.5.1 矩阵的顺序表示 .....	(53)
1.3.3 结点与结构 .....	(12)	2.5.2 稀疏矩阵的表示方法 .....	(54)
1.3.4 外存数据的组织 .....	(13)	2.6 广义表与动态存储管理 .....	(57)
1.4 算法 .....	(16)	2.6.1 广义表 .....	(58)
1.4.1 什么是算法 .....	(16)	2.6.2 结点的动态分配与回收 .....	(60)
1.4.2 算法的设计 .....	(17)	2.6.3 废料收集与存储压缩 .....	(64)
1.4.3 算法的精化 .....	(18)	小结 .....	(66)
1.4.4 算法的分析 .....	(21)	习题 .....	(66)
小结 .....	(25)		
习题 .....	(26)		
<b>第 2 章 线性表</b> .....	(29)	<b>第 3 章 字符串</b> .....	(69)
2.1 基本概念与抽象数据类型 .....	(29)	3.1 字符串及其抽象数据类型 .....	(69)
2.1.1 基本概念 .....	(29)	3.1.1 基本概念 .....	(69)
2.1.2 抽象数据类型 .....	(30)	3.1.2 抽象数据类型 .....	(70)
2.2 顺序表示 .....	(31)	3.2 字符串的实现 .....	(71)
2.2.1 存储结构 .....	(31)	3.2.1 顺序表示 .....	(71)
2.2.2 运算的实现 .....	(33)	3.2.2 链接表示 .....	(72)
2.2.3 分析与评价 .....	(36)	3.3 模式匹配 .....	(75)
2.2.4 顺序表空间的扩展 .....	(38)	3.3.1 朴素的模式匹配 .....	(75)
		3.3.2 无回溯的模式匹配 .....	(77)
		小结 .....	(83)
		习题 .....	(84)

---

<b>第 4 章 栈与队列</b> .....	(85)	5.4.2 哈夫曼树及其应用	.....	(145)	
4.1 栈及其抽象数据类型	.....	(85)	5.5 树及其抽象数据类型	.....	(151)
4.1.1 基本概念	.....	(85)	5.5.1 基本概念	.....	(152)
4.1.2 抽象数据类型	.....	(86)	5.5.2 抽象数据类型	.....	(153)
4.2 栈的实现	.....	(86)	5.5.3 树的周游	.....	(154)
4.2.1 顺序表示	.....	(86)	5.6 树的实现	.....	(157)
4.2.2 链接表示	.....	(89)	5.6.1 父指针表示法	.....	(157)
4.3 栈的应用	.....	(91)	5.6.2 子表表示法	.....	(159)
4.3.1 栈与递归	.....	(92)	5.6.3 长子 - 兄弟表示法	.....	(161)
4.3.2 迷宫问题	.....	(96)	5.6.4 树的其他表示法	.....	(162)
4.3.3 表达式计算	.....	(100)	5.7 树林	.....	(163)
4.4 队列及其抽象数据类型	.....	(102)	5.7.1 树林的周游	.....	(163)
4.4.1 基本概念	.....	(102)	5.7.2 树林的存储表示	.....	(163)
4.4.2 抽象数据类型	.....	(103)	5.7.3 树林与二叉树的转换	.....	(164)
4.5 队列的实现	.....	(103)	小结	.....	(167)
4.5.1 顺序表示	.....	(103)	习题	.....	(167)
4.5.2 链接表示	.....	(107)			
4.6 队列的应用	.....	(109)	<b>第 6 章 集合与字典</b> .....	(170)	
小结	.....	(113)	6.1 集合及其抽象数据类型	.....	(170)
习题	.....	(115)	6.1.1 基本概念	.....	(170)
			6.1.2 主要运算	.....	(171)
			6.1.3 抽象数据类型	.....	(172)
			6.2 集合的实现	.....	(173)
			6.2.1 集合的位向量表示	.....	(173)
			6.2.2 集合的单链表表示	.....	(178)
			6.3 字典及其抽象数据类型	.....	(181)
			6.3.1 基本概念	.....	(181)
			6.3.2 抽象数据类型	.....	(182)
			6.4 字典的顺序表示	.....	(182)
			6.4.1 存储结构	.....	(183)
			6.4.2 算法的实现	.....	(183)
			6.4.3 有序顺序表与二分法检索	.....	(184)
			6.5 字典的散列表示	.....	(187)
			6.5.1 基本概念	.....	(188)
			6.5.2 散列函数	.....	(188)
			6.5.3 碰撞的处理	.....	(190)
			6.5.4 散列文件	.....	(196)
			小结	.....	(198)

---

习题 .....	(199)	8.3.2 堆排序 .....	(262)
<b>第7章 高级字典结构</b> .....	(204)	8.4 交换排序 .....	(266)
7.1 字典与索引 .....	(204)	8.4.1 起泡排序 .....	(266)
7.1.1 字典的索引 .....	(204)	8.4.2 快速排序 .....	(268)
7.1.2 索引的抽象 .....	(206)	8.5 分配排序 .....	(270)
7.2 字符树 .....	(206)	8.5.1 概述 .....	(271)
7.2.1 双链树表示 .....	(207)	8.5.2 基数排序 .....	(271)
7.2.2 多链表示 .....	(207)	8.6 归并排序 .....	(274)
7.3 二叉排序树 .....	(209)	8.6.1 内排序 .....	(274)
7.3.1 二叉排序树 .....	(209)	8.6.2 外排序 .....	(277)
7.3.2 二叉排序树的检索 .....	(210)	小结 .....	(283)
7.3.3 二叉排序树的插入和构造 .....	(210)	习题 .....	(285)
7.3.4 二叉排序树的删除 .....	(213)	<b>第9章 图</b> .....	(287)
7.4 最佳二叉排序树 .....	(216)	9.1 基本概念及其抽象数据类型 .....	(287)
7.4.1 基本概念 .....	(216)	9.1.1 基本概念 .....	(287)
7.4.2 等概率的检索 .....	(218)	9.1.2 抽象数据类型 .....	(291)
7.4.3 不等概的情况 .....	(219)	9.2 图的周游 .....	(292)
7.5 平衡二叉排序树 .....	(225)	9.2.1 深度优先周游 .....	(293)
7.5.1 基本概念 .....	(225)	9.2.2 广度优先周游 .....	(294)
7.5.2 调整平衡的模式 .....	(227)	9.3 存储表示 .....	(295)
7.5.3 实现 .....	(232)	9.3.1 邻接矩阵表示法 .....	(296)
7.6 索引文件 .....	(237)	9.3.2 邻接表表示法 .....	(298)
7.6.1 多分树 .....	(237)	9.3.3 两种表示的比较 .....	(299)
7.6.2 B 树 .....	(239)	9.4 最小生成树 .....	(301)
7.6.3 B+ 树 .....	(244)	9.4.1 最小生成树及其性质 .....	(302)
小结 .....	(247)	9.4.2 最小生成树的构造 .....	(303)
习题 .....	(248)	9.5 最短路径 .....	(309)
<b>第8章 排序</b> .....	(252)	9.5.1 Dijkstra 算法 .....	(309)
8.1 基本概念 .....	(252)	9.5.2 Floyd 算法 .....	(312)
8.2 插入排序 .....	(253)	9.6 拓扑排序 .....	(315)
8.2.1 直接插入排序 .....	(253)	9.6.1 AOV 网 .....	(316)
8.2.2 二分法插入排序 .....	(255)	9.6.2 拓扑排序 .....	(316)
8.2.3 表插入排序 .....	(257)	9.7 关键路径 .....	(319)
8.2.4 Shell 排序 .....	(259)	9.7.1 AOE 网 .....	(320)
8.3 选择排序 .....	(261)	9.7.2 关键路径 .....	(320)
8.3.1 直接选择排序 .....	(261)	小结 .....	(325)
		习题 .....	(326)

---

<b>第 10 章 算法分析与设计</b> .....	(329)
10.1 算法分析技术 .....	(329)
10.1.1 空间代价分析 .....	(329)
10.1.2 时间代价分析 .....	(331)
10.2 算法设计技术 .....	(335)
10.2.1 分治法 .....	(335)
10.2.2 贪心法 .....	(336)
10.2.3 动态规划法 .....	(339)
10.2.4 回溯法 .....	(344)
10.2.5 分枝界限法与 0/1 背包 问题 .....	(347)
小结 .....	(352)
习题 .....	(353)

---

<b>索引</b> .....	(356)
<b>算法清单</b> .....	(363)
<b>参考文献</b> .....	(366)

# 第 1 章 绪 论

---

“算法与数据结构”是计算机类专业的基础课程。通过该课程的学习,可以使读者较全面地理解算法和数据结构的概念、掌握应用数据结构与算法的主要原理和方法、比较不同数据结构和算法的特点,提高使用计算机解决实际问题的能力。

本章首先通过一个实例讲解使用计算机处理实际问题的过程,然后对一些最基本、最重要的概念分别展开讨论。

本章重点包括:理解从问题到程序的主要过程;体会数据结构、算法和抽象数据类型在问题求解过程中的作用;了解数据结构的主要概念和分类方法;了解算法的概念和主要设计、分析方法。

通过本章的学习,使读者对全书有个整体把握,以便读者在后继各章学习时,能更好地理解它们的地位和相互关系。

## 1.1 从问题到程序

---

用计算机实现**问题求解**,实质上就是在计算机中建立一个解决问题的模型。用来表示问题或处理问题的模型可以有不同的抽象形式:容易被人理解但不太严格的需求模型;比较抽象但很精确的数学模型;容易被计算机理解或执行的实现模型。

程序是使用程序设计语言精确描述的**实现模型**,它是问题求解的一个可以在计算机上运行的模型。程序中描述的数据用来表示问题中涉及的对象,程序中描述的过程表示了对于数据的处理算法;通过接受实际问题的输入,经过程序的运行,便可以得到实际问题的一个解。

为一个实际问题建立一个正确的求解程序,通常可以分为以下几个阶段:

**分析阶段**。该阶段的任务,首先是弄清用户的需求是什么,然后设计者对用户的需求进行深入分析,使用规范说明语言(或数学语言等工具)给出系统的需求模型(或数学模型)。

**设计阶段。**该阶段的任务是建立求解系统的实现模型,重点是算法的设计和数据结构的设计。对于大型的、复杂的系统,这一阶段往往还包括抽象数据类型或者模块的设计。一般而言,设计过程需要从粗到细,经过多次精化才能完成。

**编码阶段。**该阶段的主要任务是采用适当的程序设计语言(C语言、C++语言或Java语言等)把设计阶段的成果编写成可执行的程序。

**调试和维护。**该阶段的任務包括使用足够的例子调试编写的程序,发现和排除程序代码中的错误;在计算机上执行程序,获得问题的解;也包括在系统投入运行后,解决在使用过程中发现的隐含错误和根据使用中提出的要求进行必要的维护和完善。

上述使用计算机为实际问题构建求解程序的流程图如图1.1所示。显然,与本课程关系最为密切的是设计阶段,但由于设计阶段介于分析和编码这两个阶段之间,因此讨论中也需要涉及它们。下面通过一个实例具体展开问题求解的主要过程,重点讨论问题的分析和算法与数据结构的设计等。

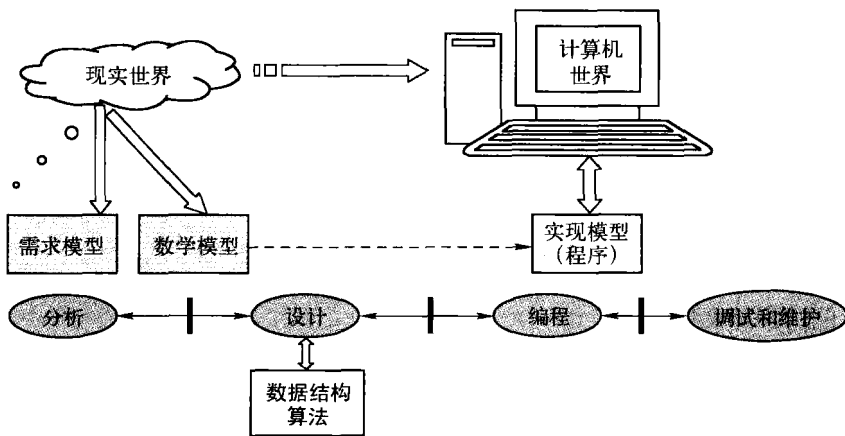


图 1.1 构建求解程序的流程图

### 1.1.1 问题分析与抽象

为了能正确地解决问题,必须首先深刻地理解需要解决的问题。只有在深刻地认识了这个问题以后,才能着手确定其解决方法。

#### 信号灯问题

考虑一个多叉路口(见图1.2),在这个路口中,共有5条道路相交,其中C和E是单行线,其他为双行线。提出的任务是:为这个路口设计一个安全有效的交通信号灯管理系统。



### 分析

为了完成上述任务,首先需要研究一下这个路口所有车辆的行驶路线,找出存在的冲突。这可以归结为对车辆的可能行驶方向做某种分组,分组的结果满足:任一个组中各个方向行驶的车辆可以同时安全行驶(不发生碰撞)。显然,对这个问题存在许多不同的分组方案,其中最简单的方案就是把每个可能的行驶路线分为一组。例如,根据这个路口的实际情况,可以确定 13 个可能通行方向: $A \rightarrow B, A \rightarrow C, A \rightarrow D, B \rightarrow A, B \rightarrow C, B \rightarrow D, D \rightarrow A, D \rightarrow B, D \rightarrow C, E \rightarrow A, E \rightarrow B, E \rightarrow C, E \rightarrow D$ 。如果把它们分在 13 个组中,每组只包含 1 种通行方向,当然非常安全。但是,这种方案是不会在实际中采用的。因为如果分组越少,可以同时行驶的车辆也就越多,路口的运行效率就越高。我们需要的就是尽可能找到一个比较高效的方案。

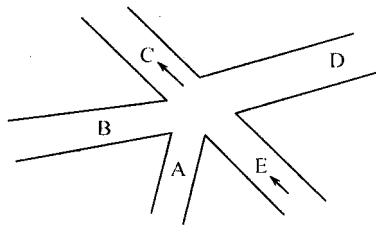


图 1.2 一个多叉路口的模型

### 抽象

根据这个路口的实际情况,在 13 个可能通行方向中,有些方向明显不能同时进行,如  $A \rightarrow B$  与  $B \rightarrow C$  等。为了叙述方便,下面把每个通行方向,诸如  $A \rightarrow B$  简写成  $AB$ ,并且用一个小椭圆把它框起来,看成一个结点;在不能同时行驶的路线间画一条连线(表示它们互相冲突),便可以得到如图 1.3 所示的网状图形。

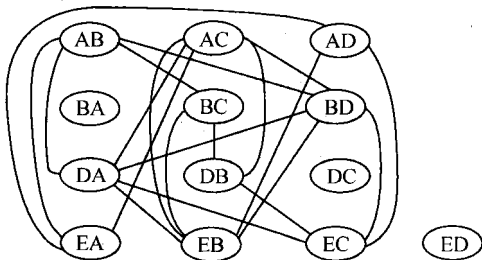


图 1.3 交叉路口行驶冲突的模型

经过上面讨论,把原来的一个实际(信号灯)问题转变成了另一个比较抽象的问题。这个问题可以借助图的模型清楚而严格地表达出来:要求将图 1.3 中的结点分组,使有线相连(互相冲突)的结点不在同一个组里。根据前面叙述的理由,我们希望对于这个问题求得一个分组数比较少的解。习惯上,把仅满足安全性的普通解称为是这个问题的一个**可行解**;把分组数最少的解称为**最优解**;而分组数接近最优解的可行解称为**次优解**。

### 着色问题

由于上面用了图的抽象表示,使得表达出的问题已经比最初提出的问题更一般、更抽象了。如果把图 1.3 中的一个结点理解为一个国家,结点之间的连线表示两国有共同边界,上述问题就是著名的**着色问题**:即如何使用尽量少的几种颜色将图中所有国家着色,使得任意两个相邻的国家颜色都不相同。解决了着色问题,也就解决了包括信号灯在内的一大类问题。