



中国科学院规划教材

数据结构

(C语言版)

陈燕 曹妍 贾红雨 李晔 编著



科学出版社

014034568

TP311.12
273

中国科学院规划教材

数据结构 (C语言版)

陈燕 曹妍 贾红雨 李晔 编著

辽宁省高等学校创新团队项目 (LT2011007) 资助出版



科学出版社

元 00.80 定价

北京



北航

C1714954

TP311.12
2273

832380110

内 容 简 介

本书是为数据结构课程编写的教材,采用逐步演算和编程运行相结合的方式,前半部分从抽象数据类型的角度,分别讨论典型数据结构的逻辑特性、存储表示和相关算法,后半部分主要讨论查找和排序的各种算法及其综合分析比较。书中对大多数算法进行时间复杂度的分析,采用C语言作为数据结构对应算法的编程语言。

本书文字通俗易懂,可作为计算机类、信息管理类、电子商务、综合管理类、信息科学类专业的本科或专科教材,或其他相关专业的选修教材,也可作为从事计算机应用等工作的科技人员的参考书。

图书在版编目(CIP)数据

数据结构: C语言版 / 陈燕等编著. —北京: 科学出版社, 2014

中国科学院规划教材

ISBN 978-7-03-040093-2

I. ①数… II. ①陈… III. ①数据结构-教材 ②C语言-程序设计-教材 IV. ①TP311.12 ②TP312

中国版本图书馆CIP数据核字(2014)第045439号

责任编辑: 马 跃 魏如萍 / 责任校对: 刘小梅

责任印制: 阎 磊 / 封面设计: 天穹教育

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>

北京市文林印务有限公司 印刷

科学出版社发行 各地新华书店经销

*

2014年3月第 一 版 开本: 787×1092 1/16

2014年3月第一次印刷 印张: 23 3/4

字数: 563 000

定价: 68.00 元

(如有印装质量问题, 我社负责调换)

本书不是一本简单的教材，而是为数据结构课程编写的教材，其内容选取符合教学大纲要求，并兼顾计算机理论及应用、计算机相关专业（学科）（如信息管理与信息系统、电子商务等）的宽泛和深层次的知识点，适用面广。

前 言

前 言

数据结构不仅是计算机科学的核心，也是计算机程序设计的重要理论技术基础和专业基础课程。随着网络与计算机技术的应用与普及，数据结构课程也逐渐成为其他理工科专业的重要选修课。本书是为数据结构课程编写的教材，其内容选取符合教学大纲要求，并兼顾计算机理论及应用、计算机相关专业（学科）（如信息管理与信息系统、电子商务等）的宽泛和深层次的知识点，适用面广。

本书的第1章综述数据、数据结构和抽象数据类型等基本概念；第2章至第7章从抽象数据类型的角度，分别讨论线性表、栈、队列、串、数组、广义表、树和二叉树以及图等基本类型的数据结构及其应用；第8章和第9章讨论查找和内部排序，除介绍各种实现方法之外，还着重从时间上进行定性或定量分析和比较。本书内容和章节编排参考清华大学严蔚敏版《数据结构（C语言版）》，沿用其中抽象数据类型的概念，对每一种数据结构都给出相应的数据结构类型的形式化定义。在此基础之上，作者结合数十年数据结构课程的教学经验，综合考虑学时、专业和学生的实际情况，进一步细化数据结构算法的讲解，同时也删去一部分内容。

本书采用逐步演算和编程运行相结合的方式，并使用C语言作为数据结构和算法的描述语言。本书对所涉及的每一种数据结构算法均给出相应的C语言实现代码，便于读者将算法的逻辑步骤与上机实现步骤进行对照，加深读者对数据结构算法的理解。特别地，针对更为复杂的算法，以第3章的Hanoi塔递归为例，本书采用图示的方式显示每一次进入递归与跳出递归时圆盘数量以及栈中参数的变化情况，在一定程度上降低理解递归算法的难度。本书可作为计算机类、信息管理类、电子商务、综合管理类、信息科学类专业专业的本科或专科教材，或其他相关专业的选修教材。本书文字通俗易懂，便于自学，也可供从事计算机应用等工作的科技人员参考。只需掌握程序设计基本技术便可学习本书。

撰写本书的目的在于使学生较全面地理解数据结构的概念、掌握各种数据结构的算法和实现方式，提高程序设计的质量，根据所求解问题的性质选择合理的数据结构并对时间、空间复杂度进行必要的控制。通过对本书的学习，能够提高学生使用计算机解决实际问题的能力。

在本书的编写过程中，王勇臻、陈冠、陈志珍、李欧、韩红云等同学参与完成部分章节中具体算法的程序实现和全书的校对工作，在此表示感谢。

本书旨在涵盖典型和有代表性的数据结构类型及其相关算法，但由于该课程覆盖的专业知识广、涉及的数学模型多，还有许多数据结构类型需要进一步探讨。在编写过程中，作者查阅了国内外大量文献资料，谨向书中提到的和参考文献中列出的学者表示感谢。由于我们工作的疏忽，可能本书中某些内容所参考的文献没有一一列出，在此向所

涉及的作者表示歉意。同时,由于时间仓促和作者能力有限,书中难免存在一些不当之处,敬请广大读者批评指正。

言 前

作 者

本书是根据作者在清华大学讲授“数据结构”课程的经验编写而成的。本书可作为高等院校计算机专业及相关专业的教材,也可供从事计算机工作的工程技术人员参考。

本书共分 11 章。第 1 章介绍数据结构的基本概念;第 2 章介绍线性表;第 3 章介绍栈和队列;第 4 章介绍树和二叉树;第 5 章介绍图;第 6 章介绍查找;第 7 章介绍排序;第 8 章介绍字符串;第 9 章介绍离散数学;第 10 章介绍算法设计;第 11 章介绍数据结构的实际应用。

本书在编写过程中参考了国内外许多优秀的教材和有关文献,并得到了许多同行和同事的帮助。在此,作者表示衷心的感谢。

由于时间仓促,书中难免存在一些不足之处,敬请广大读者批评指正。

目 录

前言

第 1 章 绪论	1
1.1 数据结构的研究与发展	1
1.1.1 国外的研究与发展	1
1.1.2 国内的研究与发展	1
1.1.3 数据结构在计算机专业中的地位	2
1.2 什么是数据结构	3
1.3 数据结构的基础知识	7
1.4 数据类型与抽象数据类型	11
1.5 算法和算法的度量	14
1.5.1 算法简述	14
1.5.2 算法的特征	14
1.5.3 算法对应的程序设计模式	15
1.5.4 时间复杂度	21
1.5.5 空间复杂度	24
1.6 数据结构的选择与评价	25
第 2 章 线性表	27
2.1 线性表的基本概念	27
2.1.1 线性表的定义	27
2.1.2 线性表的抽象数据类型定义	27
2.1.3 线性表的存储结构	29
2.1.4 线性表的抽象数据类型定义的应用	30
2.2 线性表的顺序存储结构	33
2.2.1 线性表的顺序存储结构定义	33
2.2.2 线性表的顺序存储结构的基本操作	33
2.3 线性表的链式存储结构	39
2.3.1 线性表的链式存储结构定义	39
2.3.2 线性表的链式存储结构的基本操作	40
2.3.3 循环链表与双向链式存储结构及操作	48
2.4 顺序表与链表的比较	52
2.5 线性表的应用例子	54
2.5.1 一元多项式的线性表的顺序存储结构及运算	54
2.5.2 一元多项式的线性表的链式存储结构	54

第 3 章 栈和队列	61
3.1 栈的基本概念	61
3.1.1 栈的定义	61
3.1.2 栈的抽象数据类型定义	62
3.1.3 栈的表示和实现	62
3.2 栈的应用	68
3.2.1 数制转换	68
3.2.2 括号匹配	70
3.2.3 运用栈实现行编辑程序	72
3.2.4 迷宫求解	74
3.2.5 表达式求值	81
3.3 栈与递归	88
3.3.1 递归的概念	88
3.3.2 递归过程的内部实现	90
3.3.3 递归消除	91
3.3.4 阅读一个递归程序	95
3.4 队列的基本概念	106
3.4.1 队列的定义	106
3.4.2 队列的抽象数据类型定义	107
3.4.3 队列的表示和实现	108
3.5 队列的应用——离散事件模拟的例子	115
第 4 章 串	123
4.1 串的基本概念	123
4.1.1 串的定义	123
4.1.2 串的抽象数据类型定义	124
4.1.3 C 语言函数库中的串处理函数	127
4.2 串的存储结构及算法	128
4.2.1 串的静态存储结构及算法	128
4.2.2 串的动态存储结构及算法	132
4.3 串的模式匹配算法	137
4.3.1 模式匹配的朴素算法	137
4.3.2 模式匹配的首尾匹配算法	139
4.3.3 KMP 算法	141
4.4 文本编辑的应用	147
4.4.1 文本编辑举例	147
4.4.2 高级语言程序设计的编译方法	148

第5章 数组和广义表	152
5.1 数组的基本概念	152
5.1.1 数组的定义	152
5.1.2 数组的抽象数据类型定义	153
5.1.3 数组的表示和实现	153
5.2 数组的应用——矩阵的压缩存储	158
5.2.1 特殊矩阵	158
5.2.2 稀疏矩阵	160
5.3 广义表的基本概念	177
5.3.1 广义表的定义	177
5.3.2 广义表的抽象数据类型定义	179
5.3.3 广义表的表示和实现	180
第6章 树和二叉树	185
6.1 树的基本概念	185
6.1.1 树的定义	185
6.1.2 树的抽象数据类型定义	187
6.2 二叉树的基本概念	188
6.2.1 二叉树的定义	188
6.2.2 二叉树的抽象数据类型定义	189
6.2.3 二叉树的性质	191
6.2.4 二叉树的存储结构	193
6.3 遍历二叉树	196
6.3.1 问题的提出	196
6.3.2 二叉树遍历算法	197
6.3.3 二叉树遍历递归算法的应用	202
6.4 线索二叉树	205
6.4.1 问题的提出	205
6.4.2 线索二叉树的存储结构	205
6.4.3 二叉树的中序线索化	207
6.5 树和森林	224
6.5.1 树、森林与二叉树的相互转换	224
6.5.2 树与森林的存储	226
6.5.3 树和森林的遍历	229
6.6 哈夫曼树及其应用	230
6.6.1 最优二叉树 (哈夫曼树)	230
6.6.2 哈夫曼编码	233
6.6.3 哈夫曼树与判定树	240

第 7 章 图	242
7.1 图的基本概念	242
7.1.1 图的定义	242
7.1.2 图的抽象数据类型定义	247
7.1.3 图的存储结构	248
7.2 图的遍历	253
7.2.1 深度优先搜索	254
7.2.2 广度优先搜索	266
7.3 生成树与最小生成树	271
7.3.1 Prim 算法	273
7.3.2 Kruskal 算法	276
7.3.3 生成树与图的遍历	278
7.4 两点之间的最短路径	279
7.4.1 从某个源点到其余各顶点的最短路径	280
7.4.2 每一对顶点之间的最短路径	285
7.5 拓扑排序	287
7.5.1 拓扑排序的定义	287
7.5.2 关键路径	295
第 8 章 查找	306
8.1 查找的基本概念	306
8.2 静态查找表	307
8.2.1 无序顺序表查找——顺序查找	307
8.2.2 有序顺序表的查找——折半查找	310
8.2.3 索引顺序表查找——分块查找	313
8.3 动态查找表	315
8.3.1 二叉排序树	315
8.3.2 平衡二叉树	321
8.4 哈希表	329
8.4.1 什么是哈希表	329
8.4.2 哈希函数的构造方法	330
8.4.3 处理冲突的方法	332
8.4.4 哈希表的查找	337
8.4.5 哈希表实现的比较	338
第 9 章 内部排序	340
9.1 排序的基本概念	340
9.2 插入排序	341
9.2.1 直接插入排序	341

9.2.2 折半插入排序	344
9.2.3 表插入排序	345
9.2.4 希尔排序	347
9.3 交换排序	350
9.3.1 起泡排序	350
9.3.2 快速排序	351
9.4 选择排序	354
9.4.1 简单选择排序	354
9.4.2 堆排序	356
9.5 归并排序	360
9.6 分配排序	363
9.6.1 多关键字排序	363
9.6.2 基数排序	364
9.7 各种内部排序方法的比较	368
参考文献	370

第1章 绪 论

【内容提要】 数据结构是信息管理与信息系统学科的一门专业基础课程，它为后续的专业课程提供必要的知识和技能准备。本章介绍数据结构的概念和相关术语以及数据类型的相关知识和算法分析。

【学习要求】 了解数据结构的发展历史；掌握和理解数据结构的概念；运用形式化方法定义和描述一个实际问题对应的数据结构；掌握数据结构的相关术语与基本概念；了解算法的时间复杂度与空间复杂度以及判断算法好坏的方法。

1.1 数据结构的研究与发展

1.1.1 国外的研究与发展

早在 1968 年，美国一些学校的计算机系就开始将《数据结构》作为一门计算机专业的基础课程。数据结构的课程体系最早是由美国计算机专家 D.E.Knuth 提出的，在他所著的《计算机程序设计技巧》（第一卷《基本算法》于 1968 年出版）一书中，较为系统地描述了客观世界中各类数据的计算机外部结构（逻辑结构）、计算机内部对应的存储方式（存储结构）及其形式化定义和对应的操作之间的关系。其描述方式已经与实际问题的解决方案非常贴近，数据结构作为一门计算机专业的基础课程逐渐成为现实。随后，第二卷《半数字化算法》于 1969 年出版，第三卷《排序与搜索》于 1973 年出版，这些著作中的算法及其应用，奠定了数据结构的理论基础。当时，数据结构这门课程已经被美国其他高校所接纳。后来，数据结构课程以算法为主要的讲授内容，成为当时计算机专业的一门重要的基础课程。

20 世纪 90 年代出现的 C++ 语言和 Java 语言，从根本上解决了计算机网络环境下复杂数据结构应用的问题，如 C++ 语言是面向对象的程序设计语言，它在 C 语言的基础上发展起来，但它比 C 语言更容易为人们所学习和掌握。C++ 语言以其独特的语言机制在计算机科学的各个领域得到了广泛应用。面向对象的设计思想是在原来结构化程序设计方法基础上一次质的飞跃，C++ 语言完美地体现了面向对象的各種特性。同样地，Java 语言也是如此，其优点在于：①Java 语言的风格类似于 C++ 语言；②Java 摒弃了 C++ 语言中容易引发程序错误的地方，如指针和内存管理；③Java 语言提供了丰富的类库。C++ 语言和 Java 语言跨平台、跨系统且操作方便的特点，为复杂数据结构算法的实现带来了非常灵活的操作模式。

1.1.2 国内的研究与发展

20 世纪 70 年代后期，数据结构课程在我国已经作为计算机专业中一门重要的基础与核心课程，以清华大学严蔚敏等学者为主所编制的《数据结构》教材，逻辑结构清晰、算法丰富，多年来一直作为国内多所高校数据结构课程的首选教材。清华大学的《数据

结构》教材最大的特点是：将所有形式的数据结构运用形式化定义方法进行了描述，其描述与定义方法严谨，为学生提供了一种逻辑性强的定义方法。与此同时，书中的形式化定义方法还可以将形式化定义用于其他复杂系统的数据结构的定义中。在 20 世纪 80 年代，基本上采用 FORTRAN 和 BASIC 语言来描述算法。到了 20 世纪 90 年代初期，国内大多数教材的算法采用 Pascal 描述（或类 Pascal 或伪码来描述）。如早期清华大学的《数据结构》就是采用这种描述方法。20 世纪 90 年代后期，C 语言成为数据结构算法的主要描述语言。2000 年以后，随着 C++、VC++ 版本的出现，国内的数据结构教材又以 C++、VC++ 版本为主流。之后，随着强大的网络技术的普及与应用，又出现了跨平台的数据结构版本，如 Java、.Net 版本。这些新潮语言的诞生与应用，给多系统的复杂数据的整合、定义域知识的描述提供了新的解决方案，尤其是当前大数据的形成和云计算架构的形成，给数据结构课程增添了更多新模式。

因此，数据结构是一门专业性非常强的计算机基础课程，同时，也是计算机和信息管理专业必不可缺的核心课程。换句话说，数据结构课程覆盖的专业知识广阔，涉及的解决问题的数学模型众多。随着计算机、网络技术和数据处理能力的增强，国内外开设相关专业数据结构课程的学校越来越多，且包括的内容也越来越多。由早先的线性表运算、非线性表运算、堆栈、排队、递归、树、图等传统的数据操作模式，扩充到多系统环境下的多维数据结构、复杂类型的数据结构以及跨平台模式的数据结构模式。

1.1.3 数据结构在计算机专业中的地位

数据结构不仅是计算机和信息管理专业主要的基础课程，更重要的是作为《编译原理》《操作系统》《数据库原理》《汇编语言程序设计》《管理与决策》等课程的前驱课程，该课程内容掌握程度的好坏直接反映计算机软件水平、管理与决策水平的高低。数据结构在计算机专业的地位如图 1.1 所示。

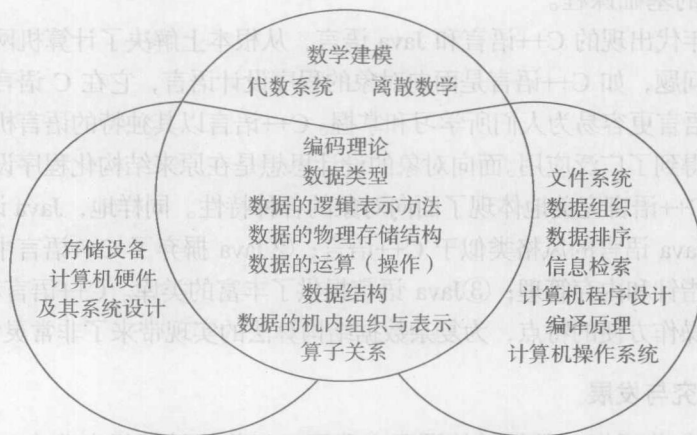


图 1.1 数据结构在计算机专业中的地位

数据结构并不是要告诉我们怎样编程，而是教会我们如何用最精练的语言和最少的资源编写出最优秀、最合理的算法与程序。换句话说，数据结构存在的意义就是使程序最优化。所以学习数据结构需要一定的基础知识。如果缺乏数据结构的知识，仅学会计算机语言的使用，想要编制出好的程序是不可能的，也不可能将客观世界复杂问题对应的实际模型进行建模和求解。同时，在大型软件开发的六个阶段（计划、需求分析、设计、编码、测试和维护）中，数据结构的内容体现在以客观管理问题求解过程中的数据建模及设计为核心，同时也涉及需求分析和编码阶段的一小部分内容。

总之，数据结构随着计算机应用与网络的普及一直在发展，并且面向特殊领域的数据结构也在不断发展，如新兴的图像处理与动画技术需要新的数据结构予以支持。面向对象技术与其他新兴技术的出现，掀起了一种从更加抽象的层次上讨论数据结构的概念和内容的新趋势。

1.2 什么是数据结构

数据结构课程主要是介绍一些常用的数据结构，阐明数据结构内在的逻辑关系，讨论它们在计算机中的存储表示和进行各种运算的实现算法。

按照字面含义，数据结构一定包括与数据及其结构相关的含义。广义上，数据结构包括数据的逻辑结构、数据的存储结构和数据运算三个方面，即涉及数据之间的逻辑关系、数据在计算机中的存储方式和相关的一组操作。

数据结构主要任务：首先，将某实际问题的逻辑结构找出来；其次，在针对该逻辑结构进行科学分析后，映射为其对应的物理结构（即数据结构）形式；最后，对该数据进行操作。

美国计算机科学家沃思教授认为：

$\text{Algorithms} + \text{Data Structure} = \text{Programs}$

Algorithm（算法）：实现程序的逻辑步骤。

Data Structure（数据结构）：描述了解决问题的数学模型；解决问题的相关数据集合；数据之间的结构关系集合。

Program（程序）：为解决实际问题而编制的一组指令集合。

算法、数据结构与程序之间的关系：

- (1) 算法是实现程序的核心；
- (2) 如果一个问题的解决方法确定，则该问题的数据结构及数据之间的运算关系也是确定的，即该数据结构的选取决定了其算法的时间复杂性——程序运行的时间和速度；
- (3) 程序运行速度与算法本身、计算机本身、操作系统、网络环境等紧密相关。

如图 1.2 所示为计算机解决问题的步骤，数据结构的工作主要体现在第 1 步“分析问题”与第 2 步“确定处理方案”，这既是最基本的工作，也是最重要的工作。下面将举

几个例子帮助理解。



图 1.2 计算机解决问题的步骤

例 1-1 求解 $Y=aX+b$ 。

具体步骤为：

- (1) 首先判断该问题是否属于计算问题，即涉及的运算对象是否属于整型、实型或布尔型的简单变量；
- (2) 运用某语言将 $Y=aX+b$ 中的 a 与 b 分别输入后，再进行计算；
- (3) 打印或显示运行结果。

例 1-2 电话号码的查询。

具体步骤为：

- (1) 构造一张电话号码登记表，表中的每个节点存放两个数据项：姓名与电话号码。
- (2) 按照链表结构构造该问题的数据存储方式，即问题对应的内部存储方式，确定链表的模式，如单链表模式如图 1.3 所示。

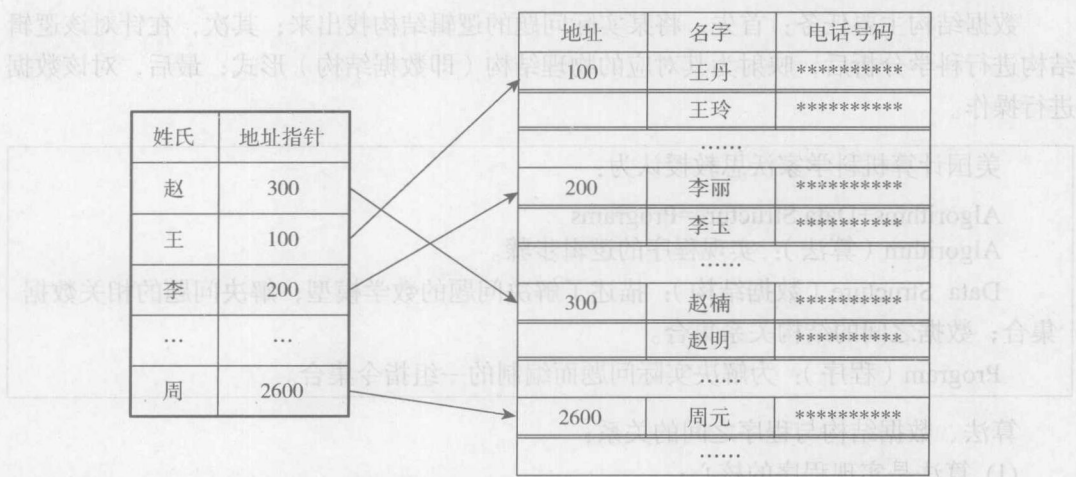


图 1.3 链式电话号码登记表

例 1-3 计算矩阵的乘积 $C[i, j]=A[i, k] \times B[k, j]$ 。

具体步骤为：

- (1) 分析问题中的各个运算参数的外部表示图 1.4。

$$A[i, k] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ip} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mp} \end{bmatrix} \quad B[k, j] = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kn} \\ \vdots & \vdots & & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pn} \end{bmatrix} \quad C[i, j] = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{i1} & c_{i2} & \cdots & c_{in} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}$$

图 1.4 矩阵运算参数的外部表示

(2) 确定 $A[i, k]$, $B[k, j]$ 与 $C[i, j]$ 矩阵的内部存储方式。

(i) 以行为主序存储。

$A[i, k]$ 在内存的顺序是 $a_{11}a_{12}\cdots a_{1p}a_{21}a_{22}\cdots a_{2p}\cdots a_{i1}a_{i2}\cdots a_{ip}\cdots a_{m1}a_{m2}\cdots a_{mp}$;

$B[k, j]$ 在内存的顺序是 $b_{11}b_{12}\cdots b_{1n}b_{21}b_{22}\cdots b_{2n}b_{i1}b_{i2}\cdots b_{in}\cdots b_{p1}b_{p2}\cdots b_{pn}$;

$C[i, j]$ 在内存的顺序是 $c_{11}c_{12}\cdots c_{1n}c_{21}c_{22}\cdots c_{2n}\cdots c_{i1}c_{i2}\cdots c_{in}\cdots c_{m1}c_{m2}\cdots c_{mn}$ 。

(ii) 以列为主序存储。

$A[i, k]$ 在内存的顺序是 $a_{11}a_{21}\cdots a_{m1}a_{12}a_{22}\cdots a_{m2}\cdots a_{i1}a_{2i}\cdots a_{mi}\cdots a_{1p}a_{2p}\cdots a_{mp}$;

$B[k, j]$ 在内存的顺序是 $b_{11}b_{21}\cdots b_{p1}b_{12}b_{22}\cdots b_{p2}\cdots b_{i1}b_{2i}\cdots b_{pi}\cdots b_{1n}b_{2n}\cdots b_{pn}$;

$C[i, j]$ 在内存的顺序是 $c_{11}c_{21}\cdots c_{m1}c_{12}c_{22}\cdots c_{m2}\cdots c_{i1}c_{2i}\cdots c_{mi}\cdots c_{1n}c_{2n}\cdots c_{mn}$ 。

(3) 寻址方式决定其计算模式。

(i) 如果以行为主序存储该矩阵乘积的寻址公式是

$\text{Loc}A[i, k] = \text{Loc}A[1, 1] + (i-1) \times p + k - 1$;

$\text{Loc}B[k, j] = \text{Loc}B[1, 1] + (k-1) \times n + j - 1$;

$\text{Loc}C[i, j] = \text{Loc}C[1, 1] + (i-1) \times n + j - 1$ 。

(ii) 如果按列为主序存储该矩阵乘积的寻址公式是

$\text{Loc}A[i, k] = \text{Loc}A[1, 1] + (k-1) \times m + i - 1$;

$\text{Loc}B[k, j] = \text{Loc}B[1, 1] + (j-1) \times p + k - 1$;

$\text{Loc}C[i, j] = \text{Loc}C[1, 1] + (j-1) \times m + i - 1$ 。

(4) 计算矩阵乘积, 显示计算结果。

例 1-4 单车道车辆进库与出库满足先进后出或者后进先出的规律, 也将满足这种规律的运算称为栈, 栈的运算与操作在栈的一端进行 (详细内容请参考本教材第 3 章栈和队列的相关内容)。

例 1-5 银行窗口业务处理, 满足先来先服务、后来后服务的排队的操作规律, 也称为队列 (详细内容请参考本教材第 3 章栈和队列的相关内容)。

例 1-6 图论问题。图的算法将在本书的后面章节介绍, 该算法主要包括图的表示方法、图的遍历、最短路径算法、影响工程进度的关键路径等算法 (详细内容请参考本教材第 7 章图的相关内容)。

多岔路口交通灯设置问题。 通常在十字路口只需要设置红、绿两种颜色的交通灯以便维持交通秩序, 但在多岔路口则需设置多种颜色的交通灯才能使车辆流量最大且保持交通秩序。如图 1.5 (a) 所示五岔路口, C 和 E 表示单行道。图 1.5 (b) 表示五岔路口的车流方向, 在路的中间有 L1~L13 共 13 个交通信号灯, 图中箭头方向表示每条路上

的车流方向,交叉和箭头同出口表示两条道路不能同时通车,否则会发生交通事故。当且仅当某个方向上的交通灯变为绿色时,此方向上的车辆才能通行。问怎样变换交通灯的颜色才能使此路口的车都能安全通过又能达到路口的最大流量?

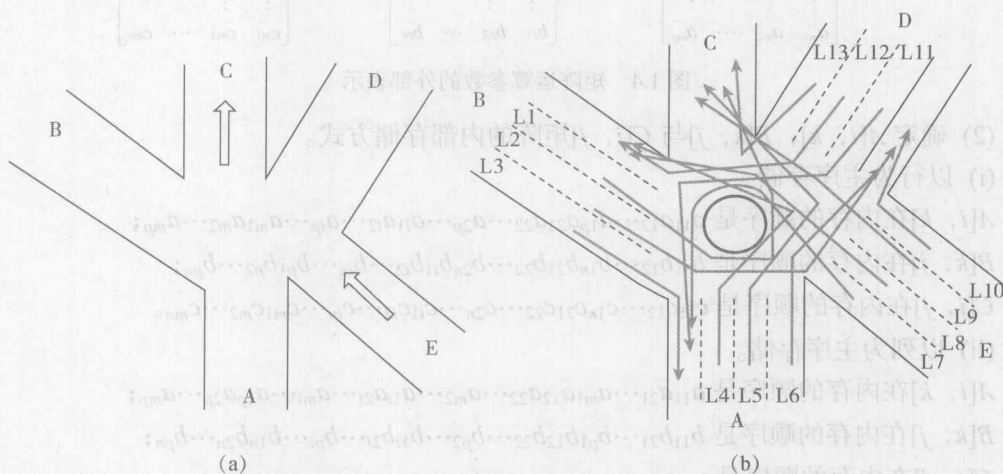


图 1.5 五岔路口交通灯设置问题

在图 1.5 中,用一个顶点表示一条通路,两条通路之间相互矛盾的关系用两个顶点之间的连线表示,即若两条通路之间不能同时通车,则两个顶点之间连一条线。因此设置交通灯的问题转换为对图中顶点的着色问题,要求对图中的每个顶点着一种颜色,并且要由线相连的两个顶点不能同时具有相同的颜色,而总的颜色种类尽可能少。

如图 1.6 所示仅为其中的一种着色方案。

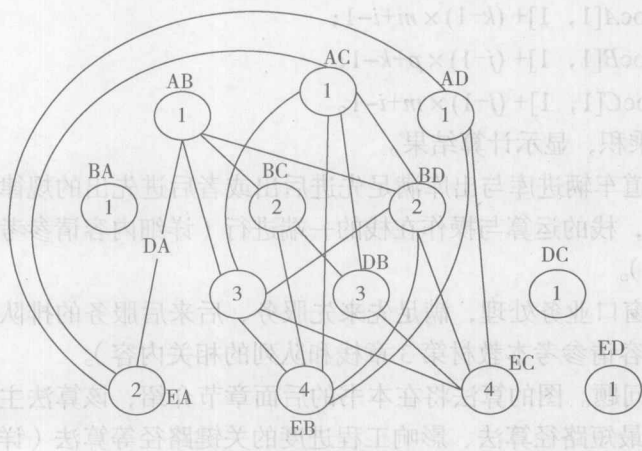


图 1.6 五岔路口交通灯设置问题一种着色方案

解决方案:

1 号灯亮时,只有 AB, BA, AC, AD, DC, ED 方向上的车能同时通过;

2 号灯亮时,只有 BC, BD, EA 方向上的车能同时通过;

3 号灯亮时,只有 DA, DB 方向上的车能同时通过;

4号灯亮时,只有EB,EC方向上的车能同时通过。

例 1-7 报文编码问题。此问题属于最优二叉树问题,即该问题对应的数据结构属于二叉树的优化问题,运用该优化解决方法解决了报文编码问题(详细内容请参考本教材第6章树的相关内容)。

另外,数据结构方面的解决方法还包括 ①将文本编辑与操作归结为字符串操作问题;②为提高查找速度,还提出各类查找算法;③为方便各类快速查找而对数据的科学排序方法。

归纳起来讲,数据结构课程是一门“描述现实世界实体的数学模型(非数值计算)及其上的操作并且在计算机中如何表示和实现”的专业基础课。

1.3 数据结构的基础知识

在本节中,我们将对一些概念和术语赋予确定的含义,以便与读者取得“共同的语言”。这些概念和术语将在以后的章节中多次出现。

数据是对客观事物的符号表示。也就是说,凡是能够输入到计算机中的,并且能够被计算机处理的符号集合都称为数据。

例如,图 1.7 中的学生成绩登记表中包括四个数据项的内容,即学号、姓名、性别与出生日期,它们其中的每一项所包含的数据分别是:学号的数据集合={0001,0002,...}、姓名的数据集合={李枫,王芳,...}、性别的数据集合={男,女}、生日的数据集合={19900507,19911125,...}。

学号	姓名	性别	出生日期
0001	李枫	男	19900507
0002	王芳	女	19911125
0005	刘平	女	19890616
0006	高山	男	19920307
...	...	...	...

图 1.7 学生登记表

数据元素是数据的基本单位。如果将某语言编制的计算机程序作为一个整体,那么,这个整体(程序)由若干个“个体”(语句)集合组成。例如,图 1.7 中的一个学生的全部信息,如{0001,李枫,男,19900507}就是一个数据元素。可以将一个数据元素分为若干个**数据项**。数据项是数据不可分割的最小单位,有时候又称为“属性”“域”或“字段”。

数据对象是性质相同的数据元素的集合,是数据的一个子集。例如,整数数据对象是集合 $N=\{0,\pm 1,\pm 2,\dots\}$;字母字符数据对象是集合 $C=\{ 'A', 'B', \dots, 'Z' \}$ 。

图 1.7 是学生登记表的例子,每个学生包括四项信息:学号、姓名、性别和出生日期。学生登记表中,整个表格是数据,每一行学生的四项信息的总和称为数据元素(数据库系统中称为记录),其中数据元素又是由四项更基本的数据构成,这些可称为数据项。

有了数据的概念之后,接下来讨论的是如何把数据送入到计算机中进行处理。数据