

21世纪高等学校本科计算机专业系列实用教材

数据结构与算法

实 用 教 程

◎ 刘玉龙 主编



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

中国计算机学会推荐教材
清华大学出版社“十二五”普通高等教育本科计算机专业规划教材

数据结构与算法

第2版

◎ 严蔚敏 吴伟英

清华大学出版社
www.tup.tsinghua.edu.cn

TP311.12

138

2007

21 世纪高等学校本科计算机专业系列实用教材

数据结构与算法实用教程

刘玉龙 主编

王 利 陈 玉 副主编

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

“数据结构与算法”是计算机科学与技术专业的一门非常重要的专业基础课,是《中国计算机科学与技术学科教程 2002》指定的核心课程之一。本书由长期担任本课程教学任务的教授主持编写,内容覆盖了该教程规定的关于本门课程的全部知识点,并融入了编者的教学经验和对课程内涵的深入思考。全书共分 10 章,内容涉及数据结构的基本概念、线性表、栈和队列、串和数组、广义表、树、图、查找、排序及算法设计方法。

本书可作为高等院校计算机专业及相关专业的教材和参考书,也可供 IT 工程技术人员参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

数据结构与算法实用教程 / 刘玉龙主编. —北京: 电子工业出版社, 2007.2

(21 世纪高等学校本科计算机专业系列实用教材)

ISBN 978-7-121-03469-5

I. 数… II. 刘… 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 011255 号

责任编辑: 刘海艳 特约编辑: 张 峻

印 刷: 北京季蜂印刷有限公司

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 18 字数: 456 千字

印 次: 2007 年 2 月第 1 次印刷

印 数: 5 000 册 定价: 26.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系电话: (010) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

编委会名单

主任委员 庄燕滨

副主任委员 张永常 邵晓根 范剑波 沈振平 倪 伟 马正华 范兴南
华容茂

委 员（以姓名笔画为序）

丁志云	丁海军	王 琳	石敏辉	刘玉龙	刘红玲	朱宇光
朱信诚	冷英男	闵立清	吴 胜	杨玉东	杨茂云	张宗杰
张碧霞	张献忠	查志琴	赵立江	赵 梅	郭小荟	徐煜明
唐土生	唐学忠	程红林	彭 珠	韩 雁		

序 言

21 世纪是“信息”主导的世纪，是崇尚“创新与个性”发展的时代，体现“以人为本”、构建“和谐社会”是社会发展的主流。然而随着全球经济一体化进程的不断推进，市场与人才的竞争日趋激烈。对于国家倡导发展的 IT 产业，需要培养大量的、适应经济和科技发展的计算机人才。

众所周知，近年来，一些用人单位对部分大学毕业生到了工作岗位后，需要 1~2 年甚至多年的训练才能胜任工作的“半成品”现象反映强烈。从中反映出单位对人才的需求越来越讲究实用，社会要求学校培养学生的标准应该和社会实际需求的标准相统一。对于 IT 业界来讲，一方面需要一定的科研创新型人才，从事高端的技术研究，占领技术发展的高地；另一方面，更需要计算机工程应用、技术应用及各类服务实施人才，这些人才可统称“应用型”人才。

应用型本科教育，简单地讲就是培养高层次应用型人才的本科学教育。其培养目标应是面向社会的高新技术产业，培养在工业、工程领域的生产、建设、管理、服务等第一线岗位，直接从事解决实际问题、维持工作正常运行的高等技术应用型人才。这种人才，一方面掌握某一技术学科的基本知识和基本技能，另一方面又具有较强的解决实际问题的基本能力，他们常常是复合性、综合性人才，受过较为完整的、系统的、有行业应用背景的“职业”项目训练，其最大的特色就是有较强的专业理论基础支撑，能快速地适应职业岗位并发挥作用。因此，可以说“应用型人才”培养既有本科人才培养的一般要求，又有强化岗位能力的内涵，它是在本科基础之上的以“工程师”层次培养为主的人才培养体系，人才培养模式必须吸取一般本科教育和职业教育的长处，兼蓄并顾。“计算机科学与技术”专业教学指导委员会已经在研究并指导实施计算机人才的“分类”培养，这需要我们转变传统的教育模式和教学方法，明确人才培养目标，构建课程体系，在保证“基础的前提”下，重视素质的养成，突出“工程性”、“技术应用性”、“适应性”概念，突出知识的应用能力、专业技术应用能力、工程实践能力、组织协调能力、创新能力和创业精神，较好地体现与实施人才培养过程的“传授知识，训练能力，培养素质”三者的有机统一。

在规划本套教材的编写时，我们遵循专业教学委员会的要求，针对“计算机工程”、“软件工程”、“信息技术”专业方向，以课群为单位选择部分主要课程，以计算机应用型人才培养为宗旨，确定编写体系，并提出以下的编写原则。

(1) 本科平台：必须遵循专业基本规范，按照“计算机科学与技术”专业教学指导委员会的要求构建课程体系，覆盖课程教学知识点。

(2) 工程理念：在教材体系编写时，要贯穿“系统”、“规范”、“项目”、“协作”等工程理念，内容取舍上以“工程背景”、“项目应用”为原则，尽量增加一些实例教学。

(3) 能力强化：教学内容的举例，结合应用实际，力争有针对性；每本教材要安排课程实践教学指导，在课程实践环节的安排上，要统筹考虑，提供面向现场的设计性、综合性的实践教学指导内容。

(4) 国际视野：本套教材的编写要做到兼长并蓄，吸收国内、国外优秀教材的特点，人才培养要有国际背景和视野。

本套教材的编委会成员及每本教材的主编都有着丰富的教学经验，从事过相关的工程项目（软件开发）的规划、组织与实施，希望本套教材的出版能为我国的计算机应用型人才的培养尽一点微薄之力。

编委会

前 言

“数据结构”是计算机学科的核心基础课程，它担负着为后继课程预备理论基础、培养学生的程序设计能力、提高学生的算法修养的重要任务，其教学质量对计算机专业学生的培养质量至关重要，对于以培养应用型 IT 人才为目标的计算机应用或信息技术专业更是如此。然而随着计算机科学技术的不断发展和应用领域的不断扩大，计算机所面对的数据结构已变得十分复杂且形式多样，数据量也呈海量之巨。如何根据各类实际问题归纳、抽象出对象的数据特征及对象间的相互联系，从而合理选择数据的组织形式和存储方法，设计出高效实用的求解算法，成为数据结构课程面临的最迫切的任务。编写一本适应上述任务的、理论上够用、偏重于应用的教材是我们的初衷和主旨，在有限的篇幅内让读者能接触到尽可能多的算法是本书追求的目标。

本书采用通俗易懂、由浅入深和循序渐进的方式介绍了数据结构的基本知识。每当引入一种新的数据结构时，在给出其抽象数据类型（ADT）之后，其对应的存储结构及基本操作算法均以 C 语言形式给出，读者可以通过上机实验来理解和验证课程的具体内容和算法过程。对于各种算法的介绍，偏重于介绍和剖析算法的基本思想、主要结构和性能优劣，而较少拘泥于对算法的时空复杂性的理论分析。

全书分为 10 章：第 1 章是绪论，介绍了数据结构的基本概念和术语，解释了解题的一般过程，以及在这一过程中抽象数据类型的作用，还介绍了算法分析的基本方法；第 2 章介绍了线性表的有关概念及其基本操作，该章是后续三章的基础；第 3 章讨论了操作受限的线性表——栈与队列的特点；第 4 章简要给出了串的处理方法；第 5 章介绍了一种常用数据类型——数组在计算机内部的表示和实现，同时介绍了广义表的概念；第 6 章介绍了非线性数据结构——树，它是典型的层次基本结构；第 7 章是关于图的内容，包括有向图和无向图；第 8 章讨论了查找的方法和技术；第 9 章介绍了常用的排序算法，包括内部排序和外部排序；第 10 章介绍了常用的算法设计方法，包括每种方法的基本思想和应用实例。

本书由长期担任本课程教学任务的教授主持编写，教材内容和编排融入了编者的教学经验和对课程特点的深入思考，具有易于使用和便于自学的特点。

本书可作为高等院校计算机科学与技术专业《数据结构与算法》课程的教材和参考书，也可供计算机工程技术人员参考，对于不设置《算法设计与分析》课程的计算机专业尤为合适。对于计算机科学与技术专业，本书可讲授 72 学时左右，去掉带星号的章节，可讲授 54 学时左右，教师可根据本校教学计划和培养目标对教学内容进行裁剪。

本书的第 1、9、10 章由刘玉龙编写，第 2、3、5 章由王利编写，第 4 章由张宗杰编写，第 6、7、8 章由陈玉编写，全书由刘玉龙统稿，刘啸编制了本书的电子教案。

本书在编写过程中得到了徐州师范大学计算机学院、宁波工程学院电信学院、苏州科技学院电子系、常州工学院计算机学院的有关领导和教师的支持与帮助，在此表示感谢。另外，本书参阅了国内外大量同类教材，在此谨向有关作者深表谢意。

由于编者水平所限，加之计算机科学技术的发展十分迅速，书中难免有错漏和不妥之处，恳请广大读者和同行赐教指正（E-mail:ylliu@xznu.edu.cn.）。

编 者

目 录

第 1 章 绪论	1
1.1 引言	1
1.1.1 解决问题的步骤	1
1.1.2 一个例子	2
1.2 数据结构	7
1.2.1 有关概念和术语	7
1.2.2 抽象数据类型	9
1.2.3 描述工具——类 C 语言	10
1.3 算法和算法分析	13
1.3.1 算法的定义及算法设计的要求	13
1.3.2 算法性能分析与度量	13
1.3.3 复杂度函数的增长率	14
1.3.4 复杂度分析的例子	15
本章小结	15
习题 1	15
第 2 章 线性表	17
2.1 线性表的类型定义	17
2.1.1 线性表的概念	17
2.1.2 线性表的抽象数据类型	18
2.1.3 线性表的例子	19
2.2 线性表的顺序表示和实现	21
2.2.1 线性表的顺序表示	21
2.2.2 顺序表操作的实现	22
2.3 线性表的链式表示和实现	26
2.3.1 单链表的表示	26
2.3.2 线性链表操作的实现	28
2.4 线性表实现方法的比较	31
2.5 循环链表	32
2.6 双向链表	34
2.7 静态链表	36
* 2.8 算法设计实例——一元多项式的表示及相加	38
本章小结	41
习题 2	41
第 3 章 栈和队列	44
3.1 栈	44

3.1.1	栈的类型定义	44
3.1.2	栈的表示和实现	45
3.1.3	顺序栈和链栈的比较	49
3.2	队列	50
3.2.1	队列的类型定义	50
3.2.2	循环队列	51
3.2.3	链队——队列的链式表示和实现	55
*3.3	递归	57
3.3.1	递归的定义	57
3.3.2	递归的实现	58
3.3.3	递归和迭代	61
3.3.4	递归的消除	62
*3.4	算法设计实例	66
3.4.1	数制转换	66
3.4.2	表达式求值算法	66
本章小结		68
习题 3		68
第 4 章 串		72
4.1	串的类型定义	72
4.1.1	串的基本概念	72
4.1.2	串的抽象数据类型	73
4.2	串的表示和实现	74
4.2.1	串的顺序存储结构	74
4.2.2	串的链式存储结构	77
4.3	串的模式匹配	78
4.3.1	朴素的模式匹配算法	78
4.3.2	KMP 算法	79
*4.4	串的应用举例	84
本章小结		85
习题 4		85
第 5 章 数组和广义表		87
5.1	数组的概念及其基本操作	87
5.1.1	数组的抽象数据类型	87
5.1.2	二维数组	88
5.2	数组的顺序存储	89
5.3	矩阵的压缩存储	90
5.3.1	特殊矩阵	90
5.3.2	稀疏矩阵	92
*5.4	广义表	96
5.4.1	广义表的定义	96

5.4.2 广义表的存储结构	97
本章小结	99
习题 5	99
第 6 章 树与二叉树	101
6.1 树的概念和操作	101
6.1.1 树的定义	101
6.1.2 树的基本术语	102
6.1.3 树的基本操作	102
6.2 二叉树	103
6.2.1 二叉树的概念及操作	103
6.2.2 二叉树的性质	104
6.2.3 二叉树的存储结构	106
6.3 二叉树的遍历	108
6.3.1 二叉树的遍历	108
6.3.2 遍历算法的应用	110
6.4 线索二叉树	113
6.4.1 线索二叉树的基本概念	113
6.4.2 线索二叉树的基本操作	114
6.5 树和森林	118
6.5.1 树的存储结构	118
6.5.2 森林、树、二叉树的相互转换	120
6.5.3 树和森林的遍历	123
6.6 哈夫曼树及其应用	124
6.6.1 最优二叉树(哈夫曼树)	124
6.6.2 哈夫曼编码	125
*6.7 算法设计实例	128
本章小结	129
习题 6	129
第 7 章 图	132
7.1 图的定义和术语	132
7.1.1 图的基本概念	132
7.1.2 图的基本操作	135
7.2 图的存储结构	135
7.2.1 数组表示法	135
7.2.2 邻接表	137
7.3 图的遍历	139
7.3.1 深度优先搜索	139
7.3.2 广度优先搜索	141
7.4 图的连通性问题	142
7.4.1 图的连通分量和生成树	142

7.4.2 最小生成树	143
7.5 有向无环图及其应用	147
7.5.1 拓扑排序	148
*7.5.2 关键路径	151
7.6 最短路径	154
7.6.1 从某个源点到其他各顶点的最短路径	155
7.6.2 每一对顶点之间的最短路径	157
*7.7 网络流问题.....	158
*7.8 算法实例.....	159
本章小结	160
习题 7.....	160
第 8 章 查找	163
8.1 概述	163
8.2 线性表上的查找	164
8.2.1 顺序表上的查找	164
8.2.2 有序表上的二分查找	166
8.3 索引表上的查找	168
8.4 树表上的查找	171
8.4.1 二叉排序树	171
8.4.2 平衡二叉树	175
*8.4.3 B_树	182
*8.4.4 键树	183
8.5 哈希表	185
8.5.1 哈希表查找的基本概念	185
8.5.2 构造哈希函数的方法	186
8.5.3 哈希冲突的解决方法	188
8.5.4 哈希表的查找及分析	189
*8.6 算法设计实例.....	191
本章小结	193
习题 8.....	193
第 9 章 排序	195
9.1 概述	195
9.1.1 排序的定义	195
9.1.2 排序的分类	195
9.1.3 排序的基本操作	196
9.1.4 排序算法的性能评价	196
9.1.5 排序算法涉及的数据类型	196
9.2 插入排序	197
9.2.1 直接插入排序	197
9.2.2 折半插入排序	199

9.2.3	希尔排序	199
9.2.4	其他插入排序	201
9.3	交换排序	201
9.3.1	冒泡排序	202
9.3.2	快速排序	203
9.4	选择排序	206
9.4.1	简单选择排序	206
9.4.2	堆排序	207
9.5	归并排序	210
9.5.1	归并排序	210
9.5.2	二路归并排序	210
9.6	分配排序	211
9.6.1	多关键字排序	211
9.6.2	基数排序	212
9.7	计数排序	214
9.7.1	计数排序	214
9.7.2	计数排序算法实现	216
9.8	内部排序比较与选择	216
9.8.1	内部排序算法性能比较	217
9.8.2	内部排序算法的选择	218
*9.9	外部排序简介	219
9.9.1	磁盘文件管理	219
9.9.2	多路归并排序	220
9.9.3	二路归并排序算法	221
9.9.4	磁盘文件排序的例子	223
9.9.5	提高外部排序算法效率的途径	227
本章小结		227
习题 9		227
第 10 章	算法设计初步	229
10.1	迭代法与穷举法	229
10.1.1	迭代法	229
10.1.2	穷举法	233
10.2	递归技术与分治法	235
10.2.1	递归技术	235
10.2.2	分治法	240
10.3	回溯法	243
10.3.1	回溯法的基本思想	243
10.3.2	0-1 背包问题	243
10.3.3	旅行商问题	244
10.3.4	求解迷宫问题	246

10.4 倒推法.....	249
10.4.1 倒推法的基本思想	249
10.4.2 倒推法应用实例	249
10.5 贪心法.....	252
10.5.1 贪心法的基本思想	252
10.5.2 使用贪心法的前提	253
10.5.3 贪心法应用实例	253
10.6 分枝限界法.....	258
10.6.1 分枝限界法的基本思想	259
10.6.2 分枝限界法的应用实例	259
10.7 动态规划法.....	262
10.7.1 动态规划法的基本思想	262
10.7.2 动态规划法的基本要素	263
10.7.3 动态规划法的变形——备忘录方法.....	265
10.8 随机算法.....	266
10.8.1 随机数的产生	267
10.8.2 用随机算法计算 π 的值	267
10.8.3 蒙特卡罗积分	268
本章小结	269
习题 10	270
参考文献	272

第1章

绪 论

本章要点

本章将重点介绍数据结构及算法的有关概念和术语，主要涉及数据结构、抽象数据类型、算法的定义及特性、算法的性能分析及度量指标、复杂度函数的增长率等。这些内容是学习后续章节的基础。

1.1 引言

随着计算机科学技术的迅猛发展，计算机的应用已从传统的数值计算领域发展到各种非数值计算领域。当前，计算机应用已经深入到情报检索、企业管理、系统工程等人类社会的各个领域。计算机的应用不再局限于早期的科学计算，而是更多地应用在信息处理方面。处理的对象不再是纯粹的数值，而是扩展到字符、表格、声音、动画和图像等多媒体信息。信息的处理也不再只是单纯的计算，而是诸如信息存储、信息检索等非数值计算的处理。面对各种应用领域的处理对象，如何选择合适的表示（结构），如何有效地存储数据，以及如何有效地实现对象之间的“运算”关系，是我们必须解决的问题。传统的解决数值计算的一套理论、方法和技术已不能满足解决非数值计算问题的需要，必须建立新的理论、方法和技术体系，于是逐渐形成了“数据结构”这一学科。因此，“数据结构”是研究和解决上述问题的重要基础，是一门研究非数值计算的程序设计问题中，计算机的操作对象及它们之间的关系和操作等问题的学科。目前，“数据结构”或“数据结构与算法”课程已成为计算机类各专业的一门重要的专业基础课。

1.1.1 解决问题的步骤

使用计算机解决上述问题时，一般需经过如下几个步骤：首先从具体问题中抽象出一个适当的数学模型，然后选择或设计一个求解此数学模型的数据结构和算法，接着编写程序进行调试、测试，最后运行该程序直至得到最终结果。

1. 分析问题

问题求解的实质是寻求从输入到输出的过程。需要注意的是：对于计算机要解决的问

题，总是要受到资源的限制的，任何计算机程序只能使用有限的空间资源，并且必须在合理的时间区间内完成运行。在分析问题时必须考虑这些限制。

2. 进行抽象

抽象的本质就是抽取问题的实质，而忽略其细节，形成数学模型。有两种形式的抽象：过程抽象和数据抽象。过程抽象是面向过程的程序设计方法经常使用的抽象方法，它将整个系统的功能划分为若干个部分，强调功能完成的过程和步骤。在面向对象程序设计方法中，所使用的抽象方法是数据抽象。所谓数据抽象，是指将要处理的数据和这些数据上的操作进行捆绑，从而形成具有不同的功能和性质的抽象数据类型。

3. 确定数据结构

一旦完成了对待处理问题的抽象，就要选取或设计合适的数据结构来表示这个问题的数据及数据之间的关系。为此，需要研究各种典型的数据结构的特点和适用范围。

4. 算法设计

确定了解决问题的数据结构后，就可以进行算法设计了。解决一个问题常会有多种算法可供选择，我们总是选择最适合的算法。有时可能没有完全适用的算法，这时就需要根据具体问题的特点和所确定的数据结构设计出相应的算法。同时，需要对算法进行分析、评估，这就要求我们建立一套算法性能分析和评估的体系和方法。

5. 程序设计

对于要解决的问题，设计了算法并进行了算法分析之后，就可以选用一种计算机程序设计语言编写相应的程序，调试并运行这个程序，给出问题的解。

如果所得到的输出（解）符合问题的要求，则这个解是有效的。反之，若输出（解）不满足问题的要求，则需要重新检查，修正上述步骤，直到解决问题。

上述五个步骤中，数据结构课程主要涉及第 2、3、4 三个步骤的内容。

1.1.2 一个例子

我们用一个例子来说明上述过程。

【例 1.1】考虑一个交叉路口（见图 1.1），在这个路口中，共有五条道路相交。其中，C 和 E 是单行线，其他为双行线。试设计一个交通信号灯管理系统。

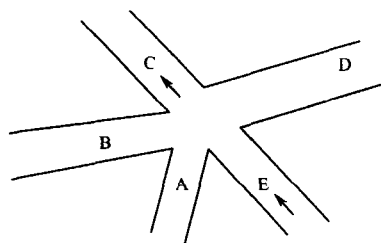


图 1.1 一个交叉路口的模型

1. 分析问题

首先需要分析一下所有车辆的行驶路线的冲突问题。这个问题可以归结为对车辆的各种可能行驶方向进行分组，每个分组对应于一种信号方案，分组的要求是要保证使任一组中各个方向行驶的车辆既可以同时安全行驶又不会发生碰撞。显然，对这个路口存在许多不同

的分组方案，且分组数越少，可以同时行驶的车辆越多，从而使管理系统的质量越高。

2. 进行抽象

根据这个路口的实际情况，可以确定 13 个可能通行的方向： $A \rightarrow B$ 、 $A \rightarrow C$ 、 $A \rightarrow D$ 、 $B \rightarrow A$ 、 $B \rightarrow C$ 、 $B \rightarrow D$ 、 $D \rightarrow A$ 、 $D \rightarrow B$ 、 $D \rightarrow C$ 、 $E \rightarrow A$ 、 $E \rightarrow B$ 、 $E \rightarrow C$ 、 $E \rightarrow D$ 。显然在右行交通系统下，有些方向不能同时进行，例如 $A \rightarrow B$ 和 $B \rightarrow C$ 这两条路线就不能同时行驶等。为了叙述方便，我们把 $A \rightarrow B$ 简写成 AB ，并且用一个称为结点的小椭圆把它框起来，在不能同时行驶的路线间画一条连线（表示它们互相冲突），结果得到如图 1.2 所示的图形。这样得到的数学模型我们称之为“图”。

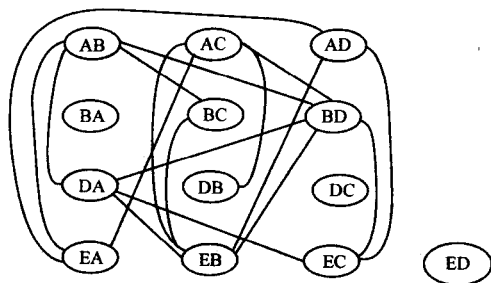


图 1.2 交叉路口的数学模型

这样就把一个实际问题变成了一个抽象问题：如何将图 1.2 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。显然，这个问题的解不是唯一的，最容易的方法就是把每个结点代表的一条行驶路线都单独分成一组，一共分成 13 组，就得到这个问题的一个“可行解”。当然，我们希望能够设计出一个可以称做“最优解”的最佳（分组数最少）的方案。

该问题可以进一步抽象为图论中的“点着色问题”，即需对图 1.2 中的结点进行着色，要求使用尽可能少的颜色进行着色，并且使得任意两个相邻的结点的颜色都不相同。

3. 确定数据结构

接下来要做的工作是：为问题中所有相关数据设计适当的表示形式，不仅包括需要表示的结点和连接，可能还有为计算过程的实现而用的辅助性数据结构。

对于本问题所涉及的图，如果有一种高级语言，它可以直接描述和处理抽象的集合和图，那么下一道工序程序设计就很容易了。此时可设需要着色的图是 G ，集合 $V1$ 包括图中所有未被着色的结点，着色开始时 $V1$ 是 G 所有结点的集合（用 $G.V1$ 表示）。用 NEW 表示已确定可以用新颜色着色的结点的集合。

关于图的典型存储结构，我们将在本书第 7 章中详细介绍。此处，不妨先选择一种所谓“邻接矩阵”的数据结构。图 1.2 所示图的邻接矩阵如图 1.3 所示。