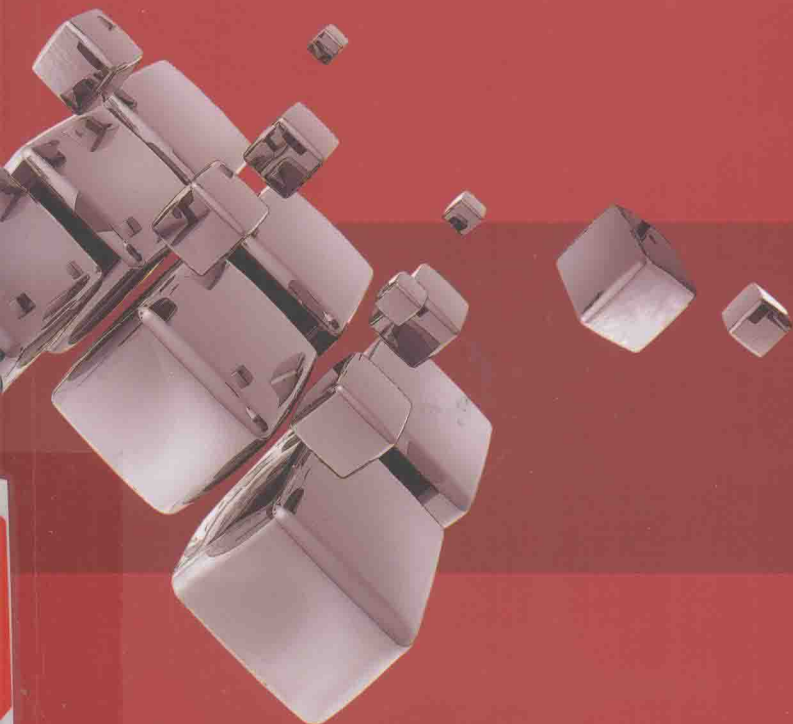


高等学校计算机规划教材

数据结构与算法

◆ 唐名华 主编
◆ 彭诗力 刘秋莲 朱海冰 韩冬 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



本书包含配套考试系统

高等学校计算机规划教材

数据结构与算法

唐名华 主 编

彭诗力 刘秋莲
编 著

朱海冰 韩 冬

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书系统地介绍了数据结构与算法的基本知识。第1章介绍了数据结构和算法相关的概念,并介绍了本书配套的考试软件的使用方法。第2章~第7章按照逻辑结构对数据结构进行了分类,具体分为线性表、栈和队列、字符串、数组和广义表、树和二叉树、图;在介绍每种数据结构的时候又按照不同的存储结构分别进行了介绍,同时介绍了各种运算在具体存储结构中的实现方法,并给出了用C语言实现的算法描述,这样就形成了逻辑结构、存储结构及运算一致的数据结构的学习思路,极其有利于初学者学习。第8章和第9章分别介绍了常用的查找和排序算法。

本书可以作为高等院校计算机相关专业的教材,也可以作为从事计算机应用开发人员的参考用书。

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 / 唐名华主编; 彭诗力等编著. —北京: 电子工业出版社, 2016.3

ISBN 978-7-121-27728-3

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

中国版本图书馆 CIP 数据核字 (2015) 第 287022 号

策划编辑: 戴晨辰

责任编辑: 郝黎明

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

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: 787×1 092 1/16 印张: 14.5 字数: 371.2 千字

版 次: 2016 年 3 月第 1 版

印 次: 2016 年 3 月第 1 次印刷

定 价: 39.00 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, 联系及邮购电话: (010) 88254888。

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

服务热线: (010) 88258888。

前 言

随着信息技术的不断发展,计算机、平板电脑、智能手机等电子设备逐渐普及,人们的生活正在悄然发生着改变。有了先进技术设备的支持,在教育领域也就能够进行比较大的改革,对传统的教学、考试手段进行比较大的革新。

传统上,在学期末进行考试有以下几方面的弊端。

(1) 临考突击。很多学生平时并不努力学习,只在临考前突击复习,考完后又把知识忘光,没有学习效果。

(2) 难以为继。一门课程的知识是前后连贯的,前面章节的知识是后面章节知识学习的基础。如果学生没有把前面的知识掌握好,到了学期的中后期,教师和学生之间就丧失了互动的基础,教学活动就难以为继。

(3) 失去良机。教师在期末考试的时候才发现学生没有学好,但为时已晚,没有机会督促学生学习。

(4) 覆水难收。一个专业的课程是一个完整的体系,低年级的课程是高年级课程的基础。如果低年级的课程没有学好,到了高年级,学生就会完全失去学习的兴趣和能力。教师们会发现高年级的课常常比较难教,一方面是因为高年级的课程本身比较难;另一方面是因为学生不具备相应的基础。

为了克服传统上进行期末考试的弊端,本书配套有一套学习及考试系统,对课程的考试形式进行了较大的改革,把期末考试改变为按章节考试,即学完一章后就进行考试,不必等到学期末再进行考试。以考试系统为基础按章节考试有以下几方面的优势。

(1) 克服传统弊端。按章节考试能够最大程度地避免期末考试的弊端。学生要想获得好成绩必须平时就学好,不能等到期末突击复习。教师能够及时发现学生学习中存在的问题,并采取措施进行整改。

(2) 维持现有秩序。有了考试系统的支持,出题、考试、改卷都可以由教师在课堂上设置,交给计算机自动完成,无需学校组织考试,不会对学校现有的教学秩序造成冲击。

(3) 减轻教师负担。出题和改卷都是考试系统自动完成的,减轻了教师的负担,教师有足够的精力来进一步提高教学质量。

(4) 提高自主能力。本学习考试系统包含 5000 余道题目,涵盖了数据结构与算法的所有知识点。以该系统为基础,学生可以进行自主学习,通过大量练习来提高其学习能力。

(5) 采用相对计分。有了考试题库的支持就可以采用相对计分。相对计分的方法如下:在相同的考试时间内,所有同学尽可能多地完成题目。完成最多题目的学生得分为 100,其他学生的得分相对于该学生计算。在相对计分的激励下,所有学生在考试的时候都不会松懈,否则其得分会受到很大的影响。此计分方法在很大程度上能够防止考试作弊的情况发生。

本书除了配套有考试系统外，还配套有算法演示软件。该软件能够演示本课程中的常用算法。它可随机生成一个数据系列，并单步演示算法的执行过程。教师能够在演示的过程中讲解算法的执行流程，学生也能够通过算法演示过程理解算法。

本书提供的所有配套资源，包括考试系统、算法演示软件及电子课件等均可登录电子工业出版社的华信教育资源网（www.hxedu.com.cn）注册后免费下载。

本书中的算法采用 C 语言描述，必要的时候辅以自然语言加以说明，方便学生阅读及调试算法。

本书第 1 章和第 8 章由唐名华编写，第 2 章和第 3 章由彭诗力编写，第 4 章和第 5 章由刘秋莲编写，第 6 章和第 7 章由朱海冰编写，第 9 章由韩冬编写。全书由唐名华统稿。

由于编者水平有限，加之时间仓促，书中难免存在缺漏和错误之处，敬请读者不吝指正。

编 者

2015 年 6 月

目 录

第 1 章 绪论	1
1.1 数据结构的基本概念	1
1.1.1 数据结构的研究对象	1
1.1.2 数据结构的研究内容	2
1.1.3 数据结构的表示方法	5
1.2 算法与算法分析	6
1.2.1 算法的概念	6
1.2.2 算法的描述方法	6
1.2.3 算法分析	7
1.2.4 常用算法设计方法	8
1.3 数据结构和算法的学习与考试软件	9
1.3.1 教师端	9
1.3.2 学生端	12
习题	13
第 2 章 线性表	15
2.1 线性表的逻辑结构	15
2.1.1 线性表的引出	15
2.1.2 线性表的逻辑结构	16
2.1.3 线性表的运算	16
2.2 线性表的顺序存储结构——顺序表	17
2.2.1 顺序表的概念	17
2.2.2 顺序表的运算	18
2.2.3 顺序表的特点	22
2.3 线性表的链式存储结构——链表	22
2.3.1 链表的概念	22
2.3.2 链表的运算	23
2.4 循环链表和双向链表	30
2.4.1 循环链表	30
2.4.2 双向链表	31
习题	33

第3章 栈和队列	35
3.1 栈	35
3.1.1 栈的基本概念	35
3.1.2 栈的顺序存储结构——顺序栈	36
3.1.3 栈的链式存储结构——链栈	38
3.2 栈的应用	40
3.2.1 表达式求值	41
3.2.2 栈与递归	43
3.3 队列	45
3.3.1 队列的基本概念	45
3.3.2 队列的顺序存储结构——顺序队列	45
3.3.3 队列的链式存储结构——链式队列	48
3.3.4 循环队列	50
3.4 队列的应用	52
习题	53
第4章 字符串	55
4.1 字符串概述	55
4.2 字符串的存储结构	56
4.2.1 字符串的顺序存储结构	56
4.2.2 字符串的链式存储结构	56
4.3 字符串的运算	57
习题	63
第5章 数组和广义表	64
5.1 数组	64
5.1.1 多维数组的顺序存储	64
5.1.2 特殊矩阵的压缩存储	65
5.2 广义表	72
5.2.1 广义表的概念	72
5.2.2 广义表的存储	73
5.2.3 广义表的运算	75
习题	76
第6章 树和二叉树	78
6.1 树	78
6.1.1 树的基本概念	78
6.1.2 树的运算	80
6.2 二叉树	80
6.2.1 二叉树的基本概念	80
6.2.2 二叉树的性质	82

6.2.3	二叉树的存储结构	83
6.2.4	二叉树的运算	85
6.3	特殊的二叉树	89
6.3.1	线索二叉树	89
6.3.2	二叉排序树	92
6.3.3	最优二叉树	100
6.3.4	堆	104
6.4	树的存储结构与运算	109
6.4.1	树的存储结构	109
6.4.2	树的运算	111
6.5	森林	114
6.5.1	森林与二叉树的转换	114
6.5.2	森林的遍历	115
	习题	115
第7章	图	116
7.1	概述	116
7.1.1	图的相关概念	116
7.1.2	图的连通性	117
7.1.3	图的基本操作	118
7.2	图的存储结构	119
7.2.1	图的邻接矩阵表示	119
7.2.2	图的邻接表表示	122
7.2.3	图的边集数组表示	129
7.2.4	图的十字链表表示	129
7.3	图的遍历	130
7.3.1	图的深度优先遍历	131
7.3.2	图的广度优先遍历	132
7.4	最小生成树	134
7.4.1	图的生成树	134
7.4.2	普里姆算法	134
7.4.3	克鲁斯卡尔算法	137
7.5	最短路径问题	140
7.5.1	单源最短路径	140
7.5.2	全源最短路径	142
7.6	有向无环图	145
7.6.1	拓扑排序	145
7.6.2	关键路径	147
	习题	150

第 8 章 查找	152
8.1 线性查找表	152
8.1.1 顺序查找	152
8.1.2 折半查找	153
8.1.3 斐波那契查找	154
8.1.4 分块查找	156
8.2 二叉排序树	157
8.2.1 二叉排序树的查找性能	157
8.2.2 平衡二叉树	158
8.3 B-树	161
8.3.1 B-树的概念	161
8.3.2 B-树的查找	162
8.3.3 B-树的插入	162
8.3.4 B-树的删除	163
8.4 哈希查找	165
8.4.1 哈希表查找	165
8.4.2 哈希函数	166
8.4.3 冲突处理	168
8.4.4 哈希查找的性能	170
习题	171
第 9 章 排序	172
9.1 基本概念	172
9.2 简单排序方法	173
9.2.1 选择排序	173
9.2.2 插入排序	174
9.2.3 冒泡排序	177
9.3 快速排序	179
9.4 堆排序	181
9.5 归并排序	183
9.5.1 归并	183
9.5.2 归并排序过程	184
9.6 基数排序	185
9.7 内部排序算法性能比较	187
9.8 外部排序	188
习题	188
附录 A 习题参考答案	189

第 1 章 绪 论

1.1 数据结构的基本概念

1.1.1 数据结构的研究对象

数据结构学科是随着程序设计技术的发展而逐渐形成和发展的。在开发程序实际问题的时候，常常用数据结构来描述问题中的数据元素及其相互间的关系。程序中采用的数据结构与程序的运行效率和处理结果有着非常密切的关系。

例 1-1：假设一个班级有 100 名学生，在计算机房进行实践课的学习时，教师需要发送文件给所有学生。通常解决这个问题有两种方法：发送广播和发送单播。

用广播方式发送的时候，教师机和学生机的关系如图 1-1 所示。

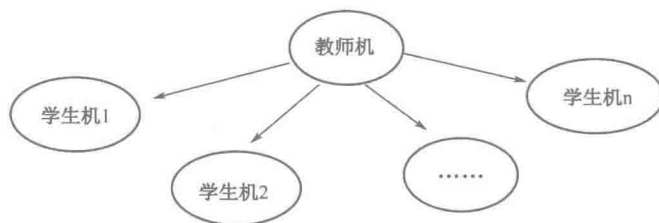


图 1-1 广播文件

用广播方式发送文件的时候，教师机发送一次，所有学生都可以收到该文件。其特点是编程容易，发送速度较快。但是，广播发送容易出现丢包的现象，学生收到的文件可能是不完整的文件。教师端必须处理这种情况，这就增加了编程的难度。同时，对丢失的数据包的处理也会降低发送速度。

用单播方式发送的时候，教师机和学生机的关系如图 1-2 所示。

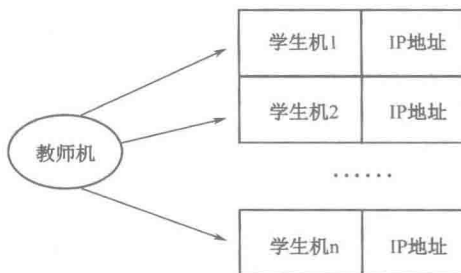


图 1-2 单播文件

用单播发送文件的时候，教师机需要知道所有学生机的 IP 地址，一次只能发送给一个学生，发送结束后，才能继续发送给下一个学生。其特点如下：保证收到，但速度较慢，

即单播发送保证每个学生都收到完整的文件，但是由于一次只能发送给一个学生，故速度较慢。

广播方式快速但不准确，单播方式准确但速度较慢。如果需要又快又准地发送文件，可以进行如下改进。

在改进的方式下，教师机和学生机之间的关系如图 1-3 所示。

教师机和每个学生机都有最多 3 个下级。教师机以单播的方式只给其 3 个下级发送文件。学生机收到文件后，也以单播方式依次给自己的 3 个下级发送文件。不同计算机给自己的下级发送文件可以同时进行。教师机先发送文件给学生机 1，然后发送文件给学生机 2；与此同时，学生机 1 可以发送文件给学生机 4。当教师机给学生机 3 发送文件的同时，学生机 1 向学生机 5 发送文件，学生机 2 向学生机 7 发送文件。

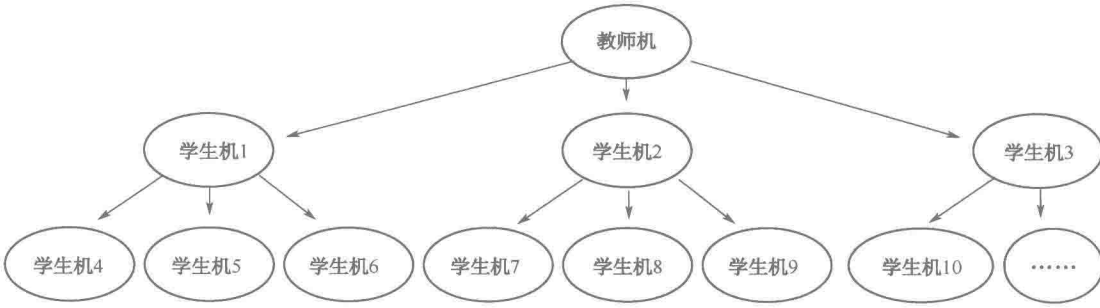


图 1-3 改进方式

在改进方式下，由于是通过单播方式发送文件的，因此可以保证每个学生机都收到完整的文件。而且由于多台计算机可以同时向不同的学生机发送文件，大大提高了发送速度。

在上面的例子中，我们分别用一个图来描述待解决的问题，用图的一个顶点来描述教师机和学生机，用顶点之间的连线来描述它们之间的关系。这就是数据结构研究的主要内容，即数据（上例中的教师机和学生机）及数据之间的关系（上例中教师机和学生机之间的关系）。从中可以看出：数据结构描述的是问题中的数据元素及其相互关系；用不同的关系把相同的一组数据元素组织起来可以得到不同的数据结构；基于不同的数据结构可以得到不同的应用程序，其效率有着巨大的差异，这就是研究数据结构的重要意义。

1.1.2 数据结构的研究内容

在计算机科学中，数据是一个比较广泛的概念，所有计算机输入、存储、处理和输出的信息都是数据，如字符、数字、图像、声音、视频等。在数据结构与算法中，数据元素是程序进行处理的一个独立的单位。数据元素可以包含若干子项目，称为数据项。在不同的问题领域，数据元素包含的数据项可能有所不同。

如需要考察期末考试后《C 语言程序设计》这门课程的成绩分布情况，数据元素可以只包含这门课程学生的分数，只有一个数据项。在图书管理系统中，数据元素是一本书的基本信息，包括书名、作者、出版社、出版时间等若干数据项。

数据结构的研究对象为问题中数据元素及其相互关系。具体来说，数据结构的研究内容包括数据的逻辑结构、数据的存储结构、数据的运算 3 个方面。

1. 数据的逻辑结构

数据的逻辑结构是指数据元素之间的逻辑关系。

例如，对一个班的学生信息按照学号从低到高的顺序排列，每个学生的信息包括学号、姓名、性别、年龄、籍贯 5 项。学生信息表如表 1-1 所示。

表 1-1 学生信息表

学号	姓名	性别	年龄	籍贯
20150101	王成	男	18	广东
20150102	李静	女	19	山东
20150103	刘华	女	20	河北
20150104	唐朝	男	18	四川
.....				

在一个大学的行政划分中，一个大学分为若干系，一个系又分为若干教研室，如图 1-4 所示，箭头所指的关系为行政单位之间的上下级关系。

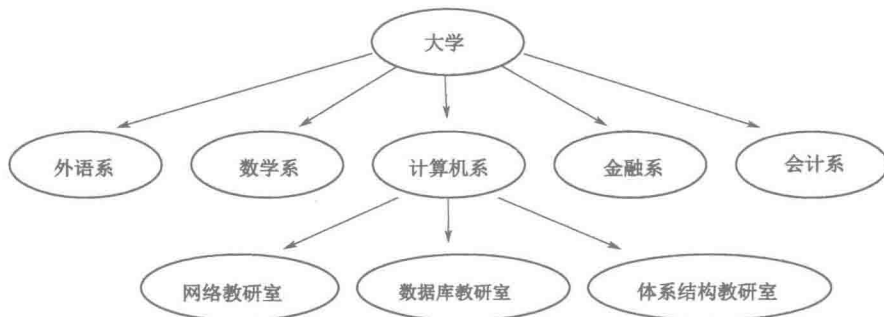


图 1-4 大学的行政划分

一个人可以有很多朋友。如果用一条线把具有朋友关系的人连接起来，就可以得到一个朋友关系网，如图 1-5 所示。

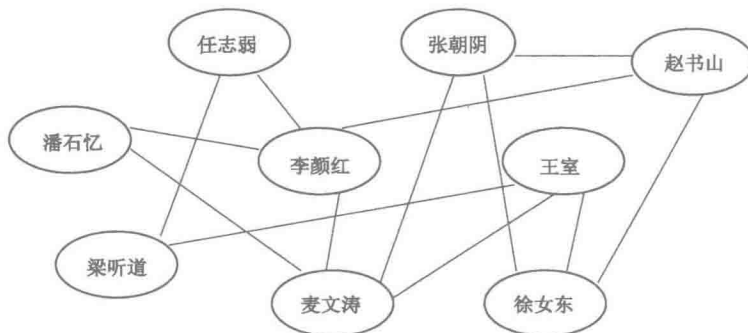


图 1-5 朋友关系网

用数据结构表示的数据元素之间的关系，有的是所研究的对象之间客观存在的，如大学行政单位之间的上下级关系、朋友关系等；有的则是为了解决问题人为加上去的，如学生机之间的上下级关系。这种关系并不一定是研究对象之间真实关系的反映，故称之为数

据的逻辑结构。

数据元素的逻辑结构可以分为两大类：线性结构和非线性结构。

在线性结构中，数据元素之间存在着一对一的关系。线性结构的特点如下。

- (1) 有且仅有一个数据元素无前驱，且只有一个后继，称为头元素。
- (2) 有且仅有一个数据元素无后继，且只有一个前驱，称为尾元素。
- (3) 其余的数据元素有且仅有一个前驱，有且仅有一个后继。

在非线性结构中，每个数据元素可能与 0 个或多个其他数据元素有关系。非线性结构又可以分为两种：树结构和图形结构。树结构如图 1-3 和图 1-4 所示，数据元素之间存在着一对多的关系。图形结构如图 1-5 所示，数据元素之间存在着多对多的关系。

2. 数据的存储结构

数据的存储结构是指数据元素及数据元素的关系在计算机中的存储（或表示），也叫做数据的物理结构。在数据的存储结构中，不仅要把数据元素存储起来，还要把数据元素之间的关系表达出来。

根据其表达数据元素之间关系的不同，数据的存储结构可以分为以下 4 种。

(1) 顺序存储。在顺序存储中，数据元素按照其逻辑结构规定的逻辑顺序，依次存储在一组连续、等长的内存单元中。数据元素之间的关系是通过存储单元的物理地址的前后顺序来表达的。

学生信息表的顺序存储如图 1-6 所示。学号相邻的学生其存储地址也相邻。

1	20150101王成 男18广东
2	20150102李静 女19山东
3	20150103刘华 女20河北
4	20150104唐朝 男18四川
...	

图 1-6 学生信息表的顺序存储

(2) 链式存储。在链式存储中，数据元素存储在任意的内存单元（可以相邻，也可以不相邻）中。数据元素之间的关系是通过指示数据元素存储地址的指针来表达的。

学生信息表的链式存储如图 1-7 所示。学号相邻的学生的存储地址不相邻，学生的逻辑关系是通过一个指示后继学生地址的指针项表达的。例如，王成的后继为李静，李静的存储地址为 1，故王成的指针项为 1，其余以此类推。

1	20150102李静 女19山东	4
2	20150104唐朝 男18四川	^
3	20150101王成 男18广东	1
4	20150103刘华 女20河北	2
...		...

H →

图 1-7 学生信息表的链式存储

(3) 散列存储。在散列存储中,数据元素的存储位置由数据元素的值确定。数据元素之间的关系通过指针表达。

(4) 索引存储。在索引存储中,需要建立索引表和子表。数据元素存储在子表中,索引表存储子表的首地址及相关信息。根据需要,索引表和子表都可以采用顺序存储或者链式存储。

散列存储和索引存储将结合查找算法进行讨论。

3. 数据的运算

数据的运算是指广义上的运算,不同的数据结构往往具有不同的运算。一般来说,常见的运算包含以下几种。

(1) 查找:查找满足给定条件的数据元素。如在学生信息表中查找学号为 20150103 的学生信息。

(2) 插入:在指定的位置加入新的数据元素。如在学生信息表的最后插入一个新学生信息。

(3) 删除:删除满足给定条件的数据元素。如在学生信息表中删除学号为 20150102 的学生信息。

(4) 修改:修改其中的某些数据元素。如在学生信息表中把学号为 20150102 的学生的年龄修改为 20。

(5) 遍历:不重复地访问所有的数据元素。

(6) 排序:按照给定的顺序输出所有的数据元素。如按照年龄从小到大的顺序输出学生信息表中的所有学生信息。

对于同一种运算,在不同的数据结构中有着不同的实现方法,且效率也有较大的差异。如在顺序存储的线性结构中实现查找运算效率较高,但是实现插入和删除运算效率较低;而在链式存储的线性结构中实现查找运算效率较低,但是实现插入和删除运算效率较高。

1.1.3 数据结构的表示方法

数据结构的表示方法有两种:二元组和图形。

在数据结构的二元组表示方法中,将数据结构形式定义为一个二元组 (D, R) , D 是数据元素的有限集合, R 是 D 上关系的有限集合。其中关系的表示方法如下:如 x 、 y 是 D 中的两个数据元素,用有序对 $\langle x, y \rangle$ 表示数据元素 x 和 y 之间的关系, x 是有序对 $\langle x, y \rangle$ 的第一个元素, y 是有序对 $\langle x, y \rangle$ 的第二个元素。

在学生信息表中,如果用 $d1$ 、 $d2$ 、 $d3$ 和 $d4$ 分别表示学号为 20150101、20150102、20150103 和 20150104 的 4 位学生,则学生信息表的二元组表示为 $\langle D, R \rangle$, 其中 $D = \{d1, d2, d3, d4\}$, $R = \{\langle d1, d2 \rangle, \langle d2, d3 \rangle, \langle d3, d4 \rangle\}$ 。

数据结构的二元组表示不是很直观。用图形表示数据结构是一种较直观的方法。在数据结构的图形表示中,用中间标有数据元素值的圆圈表示集合 D 中的元素,用带箭头的连线表示关系 R 中的有序对,箭头从有序对的第一个元素指向第二个元素。

学生信息表的图形表示如图 1-8 所示。



图 1-8 学生信息表的图形表示

假设有数据结构 $K = (D, R)$ ，其中， $D = \{a, b, c, d, e, f, g\}$ ， $R = \{ \langle a, b \rangle, \langle a, c \rangle, \langle b, d \rangle, \langle b, e \rangle, \langle c, f \rangle, \langle c, g \rangle \}$ ，则它的图形表示如图 1-9 所示。

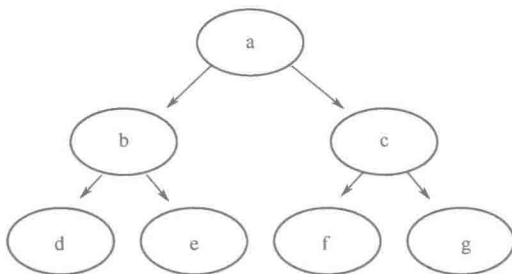


图 1-9 数据结构 K 的图形表示

用图形表示数据结构比较形象、直观，方便讨论问题，故常常被人们使用。

1.2 算法与算法分析

1.2.1 算法的概念

计算机算法是指解题步骤的精确描述。通过编写计算机程序，计算机算法最终要到计算机中运行，所以，一个计算机算法应当具有以下 5 个特征。

1) 有穷性

一个算法应当在执行有限步之后结束，不能出现无穷循环。

2) 确定性

算法中的每一个步骤必须具有确切的含义，不能使用会导致歧义的描述。同时，算法的执行路径也应当是确切的，相同的输入一定会得到相同的输出。

3) 可行性

算法中的每一个操作步骤都能够通过执行有限次已经实现的基本操作来实现。

4) 输入

算法具有零个或多个输入。输入可以由用户提供，也可以在算法中产生。

5) 输出

算法必须具有一个或多个输出。算法的运行结果应当输出给用户。

1.2.2 算法的描述方法

一个算法可以用自然语言、流程图、N-S 图和程序设计语言等方式来描述。用自然语言描述算法比较简单，人们不需要重新学习一门新的算法描述语言。但是，自然语言具有较大的歧义性，给人们交流算法和编写程序带来较大的困难。流程图和 N-S 图具有较规范的控制结构，能够准确描述算法。但是在计算机中画流程图和 N-S 图的工作量较大，而且

一旦确定就很难修改和扩充，效率比较低。程序设计语言具有规范的控制结构和较强的表达能力，可以准确地描述算法。用程序设计语言描述的算法也有利于编写程序。

本书选用功能丰富、使用灵活、应用广泛、表达能力强的 C 语言作为算法描述工具。为了节省宝贵的篇幅，本书会省略一些编译预处理命令，必要时会用一些自然语言作为辅助。

1.2.3 算法分析

对于同一个问题，常常能够设计出多种不同的算法。虽然这些算法都能够解决针对的问题，但是不同的算法常常具有较大的差别。有的算法可读性较强，有的则不然；有的算法效率较高，有的则较低。因此，有必要对算法进行分析和评价，以选出较好的算法。通常，可以从以下几个方面对算法进行评价。

1) 正确性

正确性是指对给定的一组合法输入，算法能够在有限的时间内得出预计的结果。在实际工作中，常常用测试的方法来验证算法是否满足正确性的要求。精心设计若干测试用例来运行算法，以检验算法的正确性。

2) 可读性

可读性较高的算法是易于阅读和理解的，方便人们交流，同时减少了扩充和修改的工作量。

3) 健壮性

一个健壮的算法能够对输入数据的合法性进行检查。当有非法的输入数据时，能够进行适当的处理并通知用户，不会产生一些莫名其妙的输出数据。

4) 效率

算法的效率包括时间效率和空间效率，分别称为算法的时间复杂度和空间复杂度。算法的时间复杂度越低，其运行的时间就越短。算法的空间复杂度越低，其需要的存储空间就越少。

算法的时间复杂度是指随着问题规模的增加，算法的执行时间增加的速度。算法的时间复杂度为 $O(f(n))$ ，算法的执行时间的增加速度和函数 $f(n)$ 的数量级相同，其中 n 指问题的规模。

一个算法的时间复杂度为 $O(1)$ ，是指它的执行时间不随问题规模的增加而增加。这是最理想的情况。

一个算法的时间复杂度为 $O(\log n)$ ，是指它的执行时间随问题规模的增加而以对数阶增加。这样的算法称为对数时间复杂度的算法，是性能优良的算法。

一个算法的时间复杂度为 $O(n)$ ，是指它的执行时间随问题规模的增加而线性增加。这样的算法称为线性时间复杂度的算法，也是性能优良的算法。

一个算法的时间复杂度为 $O(n \log n)$ ，是指它的执行时间随问题规模的增加而以 $n \log n$ 增加。这样的算法是性能较好的算法。

一个算法的时间复杂度为 $O(n^k)$ ($k > 1$)，它的执行时间随问题规模的增加而以多项式速度增加。这样的算法称为多项式时间复杂度的算法，要尽力降低其复杂度。

一个算法的时间复杂度为 $O(2^n)$ 或者 $O(n!)$ ，它的执行时间随问题规模的增加而以指数速度增加。这样的算法称为指数时间复杂度的算法，要避免设计这样的算法。

在分析算法的时间复杂度的时候，常以算法中重复执行的操作次数来分析。重复执行的操作次数称为该操作的语句频度。

例如，有程序段：

```
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
        a[i][j]=i+j;
```

其中，语句 $a[i][j]=i+j$ 要执行 n^2 次，故该程序段的时间复杂度为 $O(n^2)$ 。

又如，有程序段：

```
for(i=1;i<=n;i++)
    for(j=1;j<=i;j++)
        a[i][j]=i+j;
```

其中，语句 $a[i][j]=i+j$ 要执行 $n(n+1)/2$ 次，该程序段的时间复杂度也为 $O(n^2)$ 。

再如，有程序段：

```
i=1;
while(i<=n)
    i=i*2;
```

其中，语句 $i=i*2$ 要执行 $\log_2 n$ 次，故该程序段的时间复杂度为 $O(\log_2 n)$ 。

算法的空间复杂度是指随着问题规模的增加，算法需要的辅助内存空间增加的速度。与算法的时间复杂度类似，算法的空间复杂度也用记号 O 表示。如果一个算法所需要的辅助内存空间不随问题规模的增加而增加，则称该算法是原地工作的。

一个算法的时间复杂度和空间复杂度可能会相互影响。增加算法的空间开销就能够降低其运行时间，反之，为了节约空间必须延长其运行时间。在实际设计算法的时候，需要在二者之间进行权衡，找到一个平衡点，把时间和空间复杂度控制在可接受的范围内。

1.2.4 常用算法设计方法

经过长时间的摸索，针对不同类型的问题，人们总结了一些常用的算法设计方法。熟练掌握这些方法，可以加快算法设计进程。

1) 穷举法

当一个问题的解空间中解的数量不多，且随着问题规模增长而缓慢增加时，可以采用穷举法，将所有可能的解全部列出，然后选择其中较优的解。当解空间中解的数量较多时，采用穷举法会花费大量的时间，甚至不能完全列出其全部解。

2) 分治法

如果一个问题可以分解成若干子问题，且子问题与原问题结构相似，仅规模变小，则可以采用分治法求解该问题。