

新世纪计算机基础教育丛书

丛书主编 谭浩强

实用数据结构

(C++描述)(第二版)

徐士良 葛兵 编著



清华大学出版社



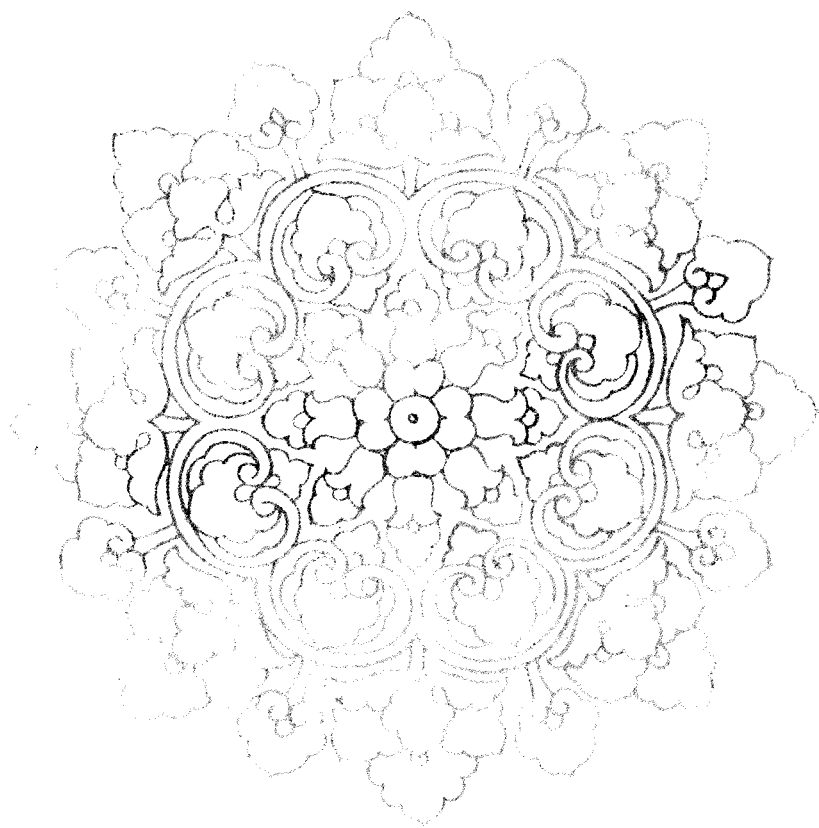
新世纪计算机基础教育丛书

丛书主编 谭浩强

实用数据结构

(C++描述)(第二版)

徐士良 葛兵 编著



清华大学出版社
北京

内 容 简 介

本书介绍了数据处理领域中常用的数据结构及其主要的运算,所有的算法均给出了 C++ 的描述,主要内容包括:数据结构与算法的基本概念、线性表及其顺序存储结构、线性链表及其运算、数组、树与二叉树、图、查找技术、Hash 表技术、排序技术。本书通俗易懂,实例丰富,每章后面都有习题。

本书可作为高等院校非计算机专业的学生学习数据结构的教材,也可作为数据结构的培训班教材或自学教材。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

实用数据结构(C++ 描述)/徐士良,葛兵编著. —2 版. —北京:清华大学出版社,2006. 11

(新世纪计算机基础教育丛书/谭浩强主编)

ISBN 7-302-13662-9

I. 实… II. ①徐… ②葛… III. 数据结构—高等学校—教材 IV. TP311.12

中国版本图书馆 CIP 数据核字 (2006) 第 097543 号

出 版 者: 清华大学出版社

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

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

组稿编辑: 焦 虹

文稿编辑: 张为民

印 刷 者: 北京国马印刷厂

装 订 者: 三河市金元印装有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 17.5 字数: 411 千字

版 次: 2006 年 11 月第 2 版 2006 年 11 月第 1 次印刷

书 号: ISBN 7-302-13662-9/TP·8247

印 数: 1~4000

定 价: 24.00 元

丛书序言

Preface Preface Preface Preface

现代科学技术的飞速发展,改变了世界,也改变了人类的生活。作为新世纪的大学生,应当站在时代发展的前列,掌握现代科学技术知识,调整自己的知识结构和能力结构,以适应社会发展的要求。新世纪需要具有丰富的现代科学知识,能够独立解决面临的任務,充满活力,有创新意识新型人才。

掌握计算机知识和应用,无疑是培养新型人才的一个重要环节。计算机技术已深入到人类生活的各个角落,与其他学科紧密结合,成为推动各学科飞速发展的有力的催化剂。无论学什么专业的学生,都必须具备计算机的基础知识和应用能力。计算机既是现代科学技术的结晶,又是大众化的工具。学习计算机知识,不仅能够掌握有关的知识,而且能培养人们的信息素养。它是高等学校全面素质教育中极为重要的一部分。

高校计算机基础教育应当遵循的理念是:面向应用需要;采用多种模式;启发自主学习;重视实践训练;加强创新意识;树立团队精神,培养信息素养。

计算机应用人才队伍由两部分人组成:一部分是计算机专业出身的计算机专业人才,他们是计算机应用人才队伍中的骨干力量;另一部分是各行各业中应用计算机的人员。这后一部分人一般并非计算机专业毕业,他们人数众多,既熟悉自己所从事的专业,又掌握计算机的应用知识,善于用计算机作为工具解决本领域中的任务。他们是计算机应用人才队伍中的基本力量。事实上,大部分应用软件都是由非计算机专业出身的计算机应用人员编制的。他们具有的这个优势是其他人难以代替的。从这个事实可以看到在非计算机专业中深入进行计算机教育的必要性。

非计算机专业中的计算机教育,无论目的、内容、教学体系、教材、教学方法等各方面都与计算机专业有很大的不同,绝不能照搬计算机专业的模式和做法。全国高等院校计算机基础教育研究会自1984年成立以来,始终不渝地探索高校计算机基础教育的特点和规律。2004年,全国高等院校计算机基础教育研究会与清华大学出版社共同推出了《中国高等院校计算机基础教育课程体系2004》(简称CFC2004),由清华大学出版社正式出版发行。

1988年起,我们根据教学实际的需要,组织编写了《计算机基础教育丛书》,邀请有丰富教学经验的专家、学者先后编写了多种教材,由清华大学出版社出版。丛书出版后,迅速受到广大高校师生的欢迎,对高等学校

的计算机基础教育起了积极的推动作用。广大读者反映这套教材定位准确,内容丰富,通俗易懂,符合大学生的特点。

1999年,根据新世纪的需要,在原有基础上组织出版了《新世纪计算机基础教育丛书》。由于内容符合需要,质量较高,被许多高校选为教材。丛书总发行量突破1000多万册,这在国内是罕见的。

最近,我们又对丛书做进一步的修订,根据发展的需要,增加了新的书目和内容。本丛书有以下特点:

(1) 内容新颖。根据21世纪的需要,重新确定丛书的内容,以符合计算机科学技术的发展和教学改革的要求。本丛书除保留了原丛书中经过实践考验且深受群众欢迎的优秀教材外,还编写了许多新的教材。在这些教材中反映了近年来迅速得到推广应用的一些计算机新技术,以后还将根据发展不断补充新的内容。

(2) 适合不同学校组织教学的需要。本丛书采用模块形式,提供了各种课程的教材,内容覆盖高校计算机基础教育的各个方面。既有供理工类专业用的,也有供文科和经济类专业用的;既有必修课的教材,也包括一些选修课的教材。各类学校都可以从中选择到合适的教材。

(3) 符合初学者的特点。本丛书针对初学者的特点,以应用为目的,以应用为出发点,强调实用性。本丛书的作者都是长期在第一线从事高校计算机基础教育的教师,对学生的基础、特点和认识规律有深入的研究,在教学实践中积累了丰富的经验。可以说,每一本教材都是他们长期教学经验的总结。在教材的写法上,既注意概念的严谨和清晰,又特别注意采用读者容易理解的方法阐明看似深奥难懂的问题,做到例题丰富,通俗易懂,便于自学。这一点是本丛书一个十分重要的特点。

(4) 采用多样化的形式。除了教材这一基本形式外,有些教材还配有习题解答和上机指导,并提供电子教案。

总之,本丛书的指导思想是内容新颖、概念清晰、实用性强、通俗易懂、教材配套。简单概括为:“新颖、清晰、实用、通俗、配套”。我们经过多年实践形成的这一套行之有效的创作风格,相信会受到广大读者的欢迎。

本丛书多年来得到各方面人士的指导、支持和帮助,尤其是得到全国高等院校计算机基础教育研究会的各位专家和各高校的老师们的支持和帮助,我们在此表示由衷的感谢。

本丛书肯定有不足之处,竭诚希望得到广大读者的批评指正。

欢迎访问谭浩强网站: <http://www.tanhoqiang.com>。

丛书主编

全国高等院校计算机基础教育研究会会长

谭浩强

数据结构是一门学科,其内容是计算机软件技术的基础。随着计算机技术的发展,数据结构的内容也在不断地更新,到目前为止,数据结构的内容已经比较系统,基本反映了计算机软件技术中最基础的知识。

本书的书名为《实用数据结构(C++描述)(第二版)》,作者认为,“实用”有两方面的含义:一是书中的内容要实用,二是内容的表现形式要实用。此次对本书的修订主要是后者。书中所有的算法均采用C++语言描述。由于C++语言既可用于面向过程的程序设计,又支持面向对象的程序设计,因此,作者在对算法进行描述时,尽量采用最合适的程序设计方法。例如,对于基本的数据结构(如顺序存储与链式存储的线性表、栈、队列等)采用面向对象的方法,将数据与运算封装成类,以便在其他应用程序中直接使用;而对于同一批数据进行同类操作的各种算法(如对线性表的各种排序方法)采用面向过程的方法,将各种不同的算法用普通函数来描述。这样,书中的所有算法就可以直接在实际应用中方便地使用。

本书实例丰富,内容精练,叙述通俗易懂,每章后面都有大量的习题,可作为高等院校非计算机专业的学生学习数据结构的教材,也可作为数据结构的培训班教材或自学教材。

由于时间紧迫与水平有限,书中难免有错误或不妥之处,恳请读者批评指正。

作 者

2006年9月

关于数据结构的书已经很多,但是有一个问题始终没有解决,即学习数据结构应具备哪些预备知识。是否一定要学习 Pascal 语言或 C 语言后才能学习数据结构?在学习数据结构前是否一定要有一些离散数学或集合论方面的基本概念与知识?学习本书所介绍的数据结构,这些预备知识都可以不要。本书的起点很低,适用的读者面很广。书中对每一种常用的数据结构都没有从抽象的定义出发,也没有涉及基本理论与计算机的专业知识。在介绍一种数据结构时,总是从实例出发,通过对实例的分析、讲解来掌握数据结构的基本概念,总结出每一种数据结构的特点及其应用。当然,为了更有效地学习数据结构,读者应已经初步学会了一种程序设计语言,但并不局限于 Pascal 语言或 C 语言。在本书中,用一种最简单的描述语言来描述对数据结构的运算,但也为了方便于学过 C 语言的读者,对每一个算法都给出了 C 语言的描述。如果读者没有学过 C 语言,可以不看 C 语言的描述,也可以通过阅读用 C 语言描述的算法顺便学习 C 语言。

本书强调实用,书中所有的算法都用 C 语言编程调试通过。

全书共分 9 章。

第 1 章从两个实例着手,介绍了数据结构的基本概念,讨论了算法方面的考虑,并给出了本书主要使用的一种算法描述语言。

第 2 章介绍了一般的线性表、栈、队列等最基本的数据结构,并讨论了它们在顺序存储结构下的主要运算,介绍了栈与队列的主要应用,最后还讨论了字符串匹配的问题。

第 3 章介绍了线性表的链式存储结构及其运算,并主要讨论了线性链表在多项式运算中的应用。

第 4 章讨论了数组的顺序存储结构、规则矩阵的压缩以及稀疏矩阵的三列二维数组表示与十字链表表示。

第 5 章介绍了树与二叉树的基本概念,并主要讨论了二叉树的性质、二叉树的存储结构及其二叉树的遍历,最后还介绍了穿线二叉树的概念、表达式线性化的过程、最优二叉树用于编码等问题。

第 6 章简要介绍了图的基本概念、图的存储方式、图的两种主要遍历方法,最后讨论了最短距离问题的求解。

第7章介绍了工程中常用的查找方法,包括顺序查找、对分查找、分块查找、二叉排序树查找以及多层索引树查找等。

第8章介绍了 Hash 表的基本概念,重点介绍了几种常用的 Hash 表。

第9章介绍了各种排序的方法,还求解了拓扑分类的问题。

由于时间紧迫与水平有限,书中难免有错误或不妥之处,恳请读者批评指正。

作 者

1999 年 11 月

目 录

Catalog Catalog Catalog Catalog



绪 论

1.1 数据结构的基本概念	1
1.1.1 两个例子	1
1.1.2 什么是数据结构	4
1.1.3 数据结构的图形表示	7
1.1.4 线性数据结构与非线性数据结构	8
1.2 算法方面的考虑	9
1.2.1 什么是算法	9
1.2.2 算法的复杂度	10
习题	12



线性表及其顺序存储结构

2.1 线性表的基本概念	14
2.1.1 什么是线性表	14
2.1.2 线性表的顺序存储结构——顺序表	15
2.1.3 顺序表的基本运算——插入与删除	16
2.1.4 顺序表类	20
2.2 栈及其应用	24
2.2.1 什么是栈	24
2.2.2 栈的顺序存储及其运算	26
2.2.3 顺序栈类	28
2.2.4 表达式的计算	31
2.3 队列及其应用	37
2.3.1 什么是队列	37
2.3.2 循环队列及其运算	37
2.3.3 循环队列类	40
2.3.4 队列的应用	44

2.4 字符串	49
2.4.1 字符串的基本概念	49
2.4.2 字符串匹配	50
习题	55



线性链表

3.1 线性链表的基本概念	57
3.1.1 线性表顺序存储的问题	57
3.1.2 线性链表的存储结构	58
3.1.3 线性链表类	62
3.1.4 带链的栈	65
3.1.5 带链的队列	68
3.2 线性链表的插入与删除	72
3.3 循环链表	77
3.4 多项式的表示与运算	80
习题	88



数组

4.1 数组的顺序存储结构	90
4.2 规则矩阵的压缩	91
4.3 一般稀疏矩阵的表示	94
4.3.1 稀疏矩阵的三列二维数组表示	95
4.3.2 稀疏矩阵的线性链表表示	108
4.3.3 十字链表	117
习题	121



树与二叉树

5.1 树	123
5.2 二叉树及其基本性质	126
5.2.1 什么是二叉树	126

5.2.2	二叉树的基本性质	127
5.2.3	满二叉树与完全二叉树	128
5.3	二叉树的遍历	129
5.4	二叉树的存储结构	131
5.4.1	二叉链表	131
5.4.2	二叉链表类	133
5.5	穿线二叉树	137
5.5.1	穿线二叉树的概念	137
5.5.2	中序穿线二叉树	137
5.5.3	前序穿线二叉树	142
5.5.4	后序穿线二叉树	145
5.6	表达式的线性化	149
5.6.1	有序树的二叉树表示	149
5.6.2	如何对表达式进行线性化	151
5.7	最优二叉树及其应用	151
5.7.1	什么是最优二叉树	151
5.7.2	最优二叉树的构造	153
5.7.3	霍夫曼编码	158
	习题	159



图

6.1	图的基本概念	161
6.2	图的存储结构	162
6.2.1	关联矩阵	162
6.2.2	求值矩阵	163
6.2.3	邻接表	164
6.2.4	邻接多重表	166
6.3	图的遍历	166
6.3.1	纵向优先搜索法	167
6.3.2	横向优先搜索法	167
6.4	图邻接表类	167
6.5	最短距离问题	173
	习题	179



查找技术

7.1	顺序查找	180
7.2	有序表的对分查找	180
7.3	分块查找	185
7.4	二叉排序树查找	186
7.4.1	二叉排序树的基本概念	187
7.4.2	二叉排序树的插入	188
7.4.3	二叉排序树的删除	190
7.4.4	二叉排序树查找	192
7.5	多层索引树查找	195
7.5.1	B ⁻ 树	195
7.5.2	B ⁺ 树	205
	习题	215



Hash 表技术

8.1	Hash 表的基本概念	217
8.1.1	直接查找技术	217
8.1.2	Hash 表	218
8.1.3	Hash 码的构造	218
8.2	几种常用的 Hash 表	220
8.2.1	线性 Hash 表	220
8.2.2	随机 Hash 表	224
8.2.3	溢出 Hash 表	229
8.2.4	拉链 Hash 表	234
8.2.5	指标 Hash 表	239
	习题	239



排序技术

9.1	互换类排序	241
9.1.1	冒泡排序	241
9.1.2	快速排序	243

9.2 插入类排序.....	246
9.2.1 简单插入排序	247
9.2.2 希尔排序	249
9.3 选择类排序.....	251
9.3.1 简单选择排序	251
9.3.2 堆排序	252
9.4 拓扑分类.....	255
9.5 其他排序方法简介.....	258
9.5.1 归并排序	258
9.5.2 基数排序	261
习题	262
 参考文献	 263

第1章 绪 论

利用计算机进行数据处理是计算机应用的一个重要领域。在进行数据处理时,实际需要处理的数据元素一般有很多,而这些大量的数据元素都需要存放在计算机中,因此,大量的数据元素按什么结构存放在计算机中,以便提高数据处理的效率,并且节省计算机的存储空间,这是进行数据处理的关键问题。显然,杂乱无章的数据是不便于处理的。而将大量的数据随意地存放在计算机中,实际上也是“自找苦吃”,对数据处理更是不利。

数据结构作为计算机的一门学科,主要研究和讨论以下3方面的问题:

- (1) 数据集合中各数据元素之间所固有的逻辑关系,即数据的逻辑结构;
- (2) 在对数据进行处理时,各数据元素在计算机中的存储关系,即数据的存储结构;
- (3) 对各种数据结构进行的运算。

讨论以上各问题的主要目的是为了提高数据处理的效率。所谓提高数据处理的效率,主要包括两个方面:一是提高数据处理的速度,二是尽量节省在数据处理过程中所占用的计算机存储空间。

本章首先通过两个简单的例子来说明不同的数据存储结构对处理效率的影响,然后介绍关于数据结构的几个基本概念,最后简要介绍在进行数据处理过程中有关算法方面的考虑。

1.1 数据结构的基本概念

1.1.1 两个例子

计算机已被广泛用于数据处理。实际问题中的各数据元素之间总是相互关联的。所谓数据处理,是指对数据集合中的各元素以各种方式进行运算,包括插入、删除、查找、更改等运算,也包括对数据元素进行分析。在数据处理领域中,建立数学模型有时并不十分重要,事实上,许多实际问题是无法表示成数学模型的。人们最感兴趣的是知道数据集合中各数据元素之间存在什么关系,为了提高处理效率,应如何组织它们,即如何表示所需要处理的数据元素。

本节将通过两个例子来说明对同一批数据采用不同的表示方法后,对处理效率的影响。

例 1.1 无序表的顺序查找与有序表的对分查找。

图 1.1 是两个子表。从图中可以看出,在这两个子表中所存放的数据元素是相同的,但它们在表中存放的顺序是不同的。在图 1.1(a)所示的表中,数据元素的存放顺序是没有规则的;而在图 1.1(b)所示的表中,数据元素是按从小到大的顺序存放的。一般称前

者为无序表,后者为有序表。

下面考虑在这两种表中进行查找的问题。

首先考虑在图 1.1(a)所示的无序表中进行查找。由于在图 1.1(a)中数据元素的存放顺序没有一定的规则,因此,要在这个表中查找某个数时,只能从第一个元素开始,逐个将表中的元素与被查数进行比较,直到表中的某个元素与被查数相等(即查找成功),或者表中所有元素与被查数都进行了比较且都不相等(即查找失败)为止,这种查找方法称为顺序查找。显然,在顺序查找中,如果被查找数在表的前部,则需要比较的次数就少;但如果被查找数在表的后部,则需要比较的次数就多。特别是当被查找数刚好是表中的第一个元素时(如被查数为 35),只需要比较一次就查找成功;但当被查找数刚好是表中最后一个元素(如被查数为 46)或表中根本没有被查数时(如被查数为 67),则需要与表中所有的元素进行比较,在这种情况下,当表很大时,顺序查找是很费时间的。虽然顺序查找法的效率比较低,但由于图 1.1(a)为无序表,没有更好的查找方法,只能用顺序查找。

现在再考虑在图 1.1(b)所示的有序表中进行查找。由于有序表中的元素是从小到大进行排列的,在查找时可以利用这个特点,以便使比较次数大大减少。在有序表中查找一个数可以如下进行:

将被查数与表中的中间这个元素进行比较:若相等,则表示查找成功,查找过程结束;若被查数大于表中的中间这个元素,则表示如果被查数在表中的话,只能在表的后半部,此时可以抛弃表的前半部而保留后半部;若被查数小于表中的中间这个元素,则表示如果被查数在表中的话,只能在表的前半部,此时可以抛弃表的后半部而保留前半部,然后对剩下的部分(前半部或后半部)再按照上述方法进行查找,这个过程一直做到在某一次的比较中相等(查找成功)或剩下的部分已空(查找失败)为止。例如,如果要在图 1.1(b)所示的有序表中查找 54,则首先与中间元素 35 进行比较,由于 78 大于 35,再与后半部分的中间元素 54 进行比较,此时相等,共比较了 2 次就查找成功。这种查找方法称为有序表的对分查找。如果采用顺序查找法,在图 1.1(a)所示的无序表中查找 54 这个元素,需要比较 9 次。

显然,在有序表的对分查找中,不论查找的是什么数,也不论要查找的数在表中有没有,都不需要与表中所有的元素进行比较,并且只需要与表中很少的元素进行比较。但需要指出的是,对分查找只适用于有序表,而对于无序表是无法进行对分查找的。

实际上,在日常工作和学习中也经常遇到对分查找。例如,当需要在词典中查找一个单词时,一般不是从第一页开始一页一页地往后找,而是考虑到词典中的各单词是以英文字母为顺序排列的,因此可以根据所查单词的第一个字母,直接翻到大概的位置,然后进行比较,根据比较结果再向前或向后翻,直到找到该单词为止。这种在词典中查单词的方法类似于对分查找。

35	16
16	21
78	29
85	33
43	35
29	43
33	46
21	54
54	78
46	85

(a) 无序表

(b) 有序表

图 1.1 数据元素存放顺序不同的两个表

由这个例子可以看出,数据元素在表中的排列顺序对查找效率是有很大影响的。

例 1.2 设有一学生情况登记表如表 1.1 所示。在表 1.1 中,每个学生的情況是以学号为顺序排列的。

表 1.1 学生情况登记表

学 号	姓 名	性 别	年 龄	成 绩
970156	张小明	男	20	86
970157	李小青	女	19	83
970158	赵 凯	男	19	70
970159	李启明	男	21	91
970160	刘 华	女	18	78
970161	曾小波	女	19	90
970162	张 军	男	18	80
970163	王 伟	男	20	65
970164	胡 涛	男	19	95
970165	周 敏	女	20	87
970166	杨雪辉	男	22	89
970167	吕永华	男	18	61
970168	梅 玲	女	17	93
970169	刘 健	男	20	75

显然,如果要在表 1.1 中查找给定学号的某学生的情况是很方便的,只要根据给定的学号就可以立即找到该学生的情况。但是,如果要在该表中查找成绩在 90 分以上的所有学生的情况,则需要从头到尾扫描全表,才能将成绩在 90 分以上的所有学生找到。在这种情况下,为了找到成绩在 90 分以上的学生情况,对于成绩在 90 分以下的所有学生情况也都要被扫描到。由此可以看出,要在表 1.1 中查找给定学号的学生情况虽然很方便,但要查找成绩在某个分数段中的学生情况时,实际上需要查看表中所有学生的成绩,其效率是很低的,尤其是当表很大时这一低效率更为突出。

为了便于查找成绩在某个分数段中的学生情况,可以将表 1.1 中所登记的学生情况进行重新组织。例如,将成绩在 90 分以上(包括 90 分)、80~89 分、70~79 分、60~69 分之间的学生情况分别登记在 4 个独立的子表中,分别如表 1.2~表 1.5 所示。现在如果要查找 90 分以上的所有学生的情况,就可以直接在表 1.2 中进行查找,从而避免了对成绩在 90 分以下的学生情况进行扫描,提高了查找效率。

表 1.2 成绩在 90 分以上的学生情况登记表

学 号	姓 名	性 别	年 龄	成 绩
970159	李启明	男	21	91
970161	曾小波	女	19	90
970164	胡 涛	男	19	95
970168	梅 玲	女	17	93

由例 1.2 可以看出,在对数据进行处理时,可以根据所做的运算不同,将数据组织成不同的形式,以便于做该种运算,从而提高数据处理的效率。

表 1.3 成绩在 80~89 分之间的学生情况登记表

学 号	姓 名	性 别	年 龄	成 绩
970156	张小明	男	20	86
970157	李小青	女	19	83
970162	张 军	男	18	80
970165	周 敏	女	20	87
970166	杨雪辉	男	22	89

表 1.4 成绩在 70~79 分之间的学生情况登记表

学 号	姓 名	性 别	年 龄	成 绩
970158	赵 凯	男	19	70
970160	刘 华	女	18	78
970169	刘 健	男	20	75

表 1.5 成绩在 60~69 分之间的学生情况登记表

学 号	姓 名	性 别	年 龄	成 绩
970163	王 伟	男	20	65
970167	吕永华	男	18	61

1.1.2 什么是数据结构

简单地说,数据结构是指相互有关联的数据元素的集合。例如,向量和矩阵就是数据结构,在这两个数据结构中,数据元素之间有着位置上的关系。又如,图书馆中的图书卡片目录,则是一个较为复杂的数据结构,对于列在各卡片上的各种书之间,可能在主题、作者等问题上相互关联,甚至一本书本身也有不同的相关成分。

数据元素具有广泛的含义。一般来说,现实世界中客观存在的一切个体都可以是数据元素。例如:

描述一年四季的季节名

春,夏,秋,冬

可以作为季节的数据元素;

表示数值的各个数

18,11,35,23,16,...

可以作为数值的数据元素;

表示家庭成员的各成员名

父亲,儿子,女儿

可以作为家庭成员的数据元素。

甚至每一个客观存在的事件,如一次演出,一次借书,一次比赛等也可以作为数据元素。

总之,在数据处理领域中,每一个需要处理的对象都可以抽象成数据元素。数据元素一般简称为元素。