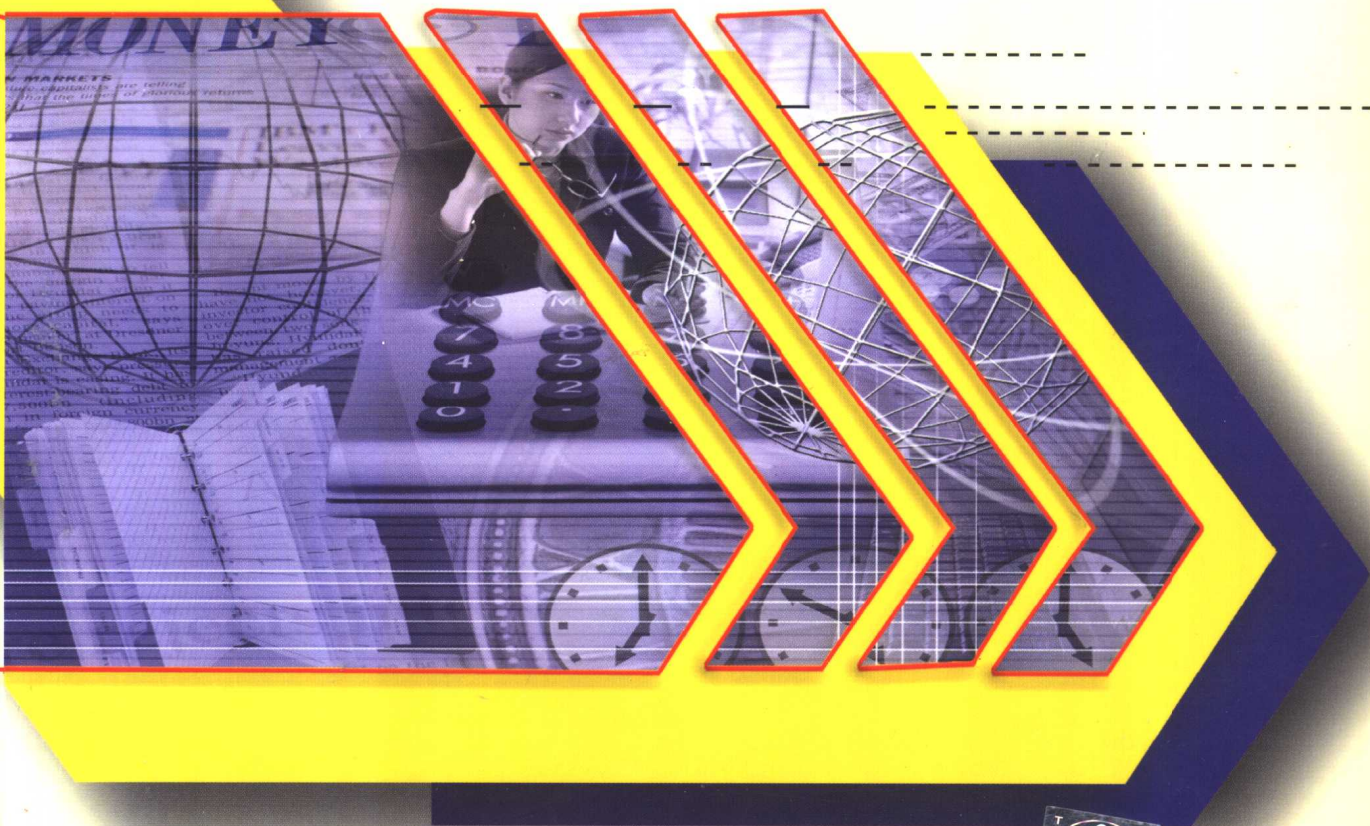


高职高专计算机系列教材

主编 谭浩强

实用数据结构 基础

陈明 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



高职高专计算机系列教材

主编 谭浩强

实用数据结构基础

陈 明 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书较系统地介绍了各种典型的数据结构,重点论述了线性表、栈和队列、递归、串、数组和广义表、树、图、查找、排序和文件。本书叙述深入浅出、选材精练、概念清楚、注重实用、逻辑性强,各章中所涉及的数据结构与算法都给出了 C 语言的描述,并附有大量的习题,便于学生理解与掌握。

本书可作为高等职业学校和大专院校的教材,也可作为计算机应用技术人员的参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

实用数据结构基础/陈明编著. —北京:清华大学出版社,2002

高职高专计算机系列教材

ISBN 7-302-05647-1

I. 实… II. 陈… III. ①数据结构—高等学校:技术学校—教材 ②C语言—程序设计—高等学校:技术学校—教材 IV. TP311.12

中国版本图书馆 CIP 数据核字(2002)第 047166 号

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 张善余

印 刷 者: 北京四季青印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 19.5 字数: 447 千字

版 次: 2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

书 号: ISBN 7-302-05647-1/TP·3329

印 数: 0001~8000

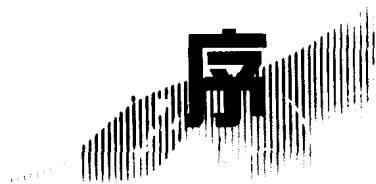
定 价: 25.00 元

编辑委员会

主 任 谭浩强
副 主 任 焦金生 陈 明 丁桂芝

委 员 (按姓氏笔画排序):

王智广	刘荫铭	朱桂兰	李文英
李 琳	李志兴	孙 慧	武绍利
张 玲	张克善	郝 玲	袁 玫
訾秀玲	薛淑斌	谢 琛	



《高职高专计算机系列教材》

到 21 世纪,计算机将成为人类的常用现代工具,每一个有文化的人都应当了解计算机,学会使用计算机,并用它来处理面临的事务。

学习计算机知识有两种不同的方法:一种是侧重知识的学习,从原理入手,注重理论和概念;另一种是侧重应用的学习,从实际入手,注重掌握其应用方法和技能。不同的人应根据其具体情况选择不同的学习方法。对多数人来说,计算机是作为一种工具来使用的,主要以应用为目的,以应用为出发点。对于高职和高专的学生,显然应当采用后一种学习方法。

传统的理论课程采用以下的三部曲:提出概念——解释概念——举例说明,这适合前面第一种方法。对于侧重应用的学习者,我们在教学实践中摸索出新的三部曲:提出问题——介绍解决问题的方法——最后归纳出一般规律或概念。实践证明这种方法是行之有效的,减少了初学者在学习上的困难。传统的方法是:先理论后实际,先抽象后具体,先一般后个别。我们采用的方法是:从实际到理论,从具体到抽象,从个别到一般,从零散到系统。我们认为,后一种方法对高职、高专和成人高教是很合适的。

本系列教材是针对高职和高专的特点组织编写的,包括了高职高专的计算机专业和非计算机专业的教材和参考书。不同专业可以从中选择所需的部分。本系列教材包含的内容比较广,除了可作为正式教材外,还可作为某些专业的选修课或指定自学的教材。

应当指出,检查学习好坏的标准,不是“知道不知道”,而是“会不会用”,学习的目的全在于应用。因此,希望读者一定要重视实践环节,多上机练习,千万不要满足于“上课能听懂、教材能看懂”。有一些问题,别人讲半天也不明白,自己一上机就清楚了。教材中有些实践性比较强的内容,不一定在课堂上由老师讲授,而应指定学生通过上机掌握。这样做可以培养学生的自学能力,启发学生的求知欲望。

本系列教材是由“浩强创作室”组织北京和天津一些普通高校和高职大学的老师们编写的,他们对高职高专的教学特点有较多的了解,有较多的实践经

验。相信本系列教材的出版会有助于高职高专的教材建设和教学改革。

由于我国的高职教育正在蓬勃发展,许多问题有待深入讨论,新的经验将会层出不穷,对如何进行高职教育将会有更新更深入的认识,本系列教材的内容也将会不断丰富和调整。我们只是为了满足许多高职高专学校对教材的急需,才下决心抓紧编写了这套系列教材,以期抛砖引玉。清华大学出版社克服了许多困难,使本系列教材在较短的时间内得以出版。

本系列教材肯定会有不足之处,请专家和读者不吝指正。

《高职高专计算机系列教材》主编
全国高等院校计算机基础教育研究会理事长
谭浩强

1999年11月1日

前 言

在非数值计算中,处理对象已从简单数值发展到具有一定结构的数据,这就需要讨论如何有效地组织计算机的存储,并在此基础上有效地实现对象间的运算,数据结构就是研究与解决这些问题的重要基础。

数据结构课是计算机科学与技术专业的一门必修的专业基础课,是计算机程序设计的重要理论技术基础。通过数据结构课的学习,不仅可以使同学们掌握数据结构的基本内容、典型算法和使用方法,而且能够训练学生应用数据结构和算法进行具体应用问题的程序设计。

本书共分 11 章,介绍最常用的数据结构、各种数据结构的逻辑关系、在计算机中的存储表示,以及在数据结构上的运算等内容,其中有线性表、栈和队列、递归、串、数组和广义表、树、图、查找、排序和文件等内容。

本书在结构上呈积木式,注重实践应用;各种常用数据结构的介绍从实际出发,避免抽象的理论论述和复杂的公式推导;在典型的算法介绍中深入浅出、简洁明了。每章都设有小结和习题。通过这些习题的练习,不仅能加深对基本概念和定义的理解,而且通过上机,能够提高编程能力和程序调试能力。

本书可以作为高等职业学校和大专院校计算机专业和相关专业的教材,也可作为从事计算机应用的工程技术人员的参考书。

由于作者水平有限,书中不足之处在所难免,敬请读者批评指正。

陈明

2001. 12. 1 于北京

目录

第1章 绪论	1
1.1 数据结构的重要性	1
1.2 基本术语	2
1.3 数据结构的概念	3
1.4 数据的逻辑结构	6
1.5 数据的存储结构	7
1.6 数据的运算	9
1.7 算法的描述和分析	9
1.7.1 算法描述	10
1.7.2 算法分析	13
1.8 小结	14
习题	14
第2章 线性表	16
2.1 线性表及逻辑结构	16
2.2 线性表的顺序存储	19
2.2.1 顺序存储	19
2.2.2 顺序结构线性表的运算	21
2.2.3 顺序存储结构的特点	24
2.3 线性表的链式存储	24
2.3.1 线性链表	24
2.3.2 线性链表的运算	28
2.3.3 循环链表	32
2.3.4 循环链表的运算	33
2.3.5 双向链表	34
2.3.6 双向链表的运算	35
2.3.7 链式存储结构的特点	37

2.4	链式存储结构的应用	37
2.4.1	约瑟夫环算法	37
2.4.2	一元多项式求和	39
2.5	小结	43
	习题	43
第3章	栈和队列	45
3.1	栈	45
3.1.1	栈的定义	45
3.1.2	栈的顺序存储结构	46
3.1.3	栈的链式存储结构	50
3.2	栈的应用	52
3.2.1	迷宫问题	52
3.2.2	算术表达式求值	55
3.3	队列	58
3.3.1	队列的定义	58
3.3.2	队列的顺序存储	60
3.3.3	队列的链式存储	66
3.4	队列的应用	71
3.5	小结	71
	习题	72
第4章	递归	74
4.1	递归的定义	74
4.2	常见递归问题	75
4.3	递归的实现	79
4.4	递归转化为非递归的一般过程	83
4.5	递归的时间和空间复杂度	87
4.6	小结	88
	习题	88
第5章	串	89
5.1	串的基本概念	89
5.2	串的存储结构	90
5.2.1	串的静态存储结构	90
5.2.2	串的动态存储结构	92
5.3	串的运算	94
5.3.1	串的基本运算	94
5.3.2	实现串的基本运算的算法	95

5.4	模式匹配	100
5.4.1	模式匹配的 BF 算法	100
5.4.2	模式匹配的 KMP 算法	103
5.5	串在文本编辑中的应用	107
5.6	小结	108
	习题	109
第 6 章	数组和广义表	111
6.1	数组的定义及其基本操作	111
6.1.1	数组的定义	111
6.1.2	数组的基本操作	112
6.2	数组的顺序存储结构	113
6.3	矩阵的压缩存储	117
6.3.1	特殊矩阵的压缩存储	117
6.3.2	稀疏矩阵的压缩存储	119
6.4	广义表的概念	133
6.5	广义表的存储结构表示	135
6.6	小结	138
	习题	138
第 7 章	树形结构	140
7.1	树	140
7.1.1	树的定义	140
7.1.2	树的表示形式	141
7.1.3	树的常用术语	142
7.1.4	树的基本操作	143
7.2	二叉树	143
7.2.1	二叉树的定义	143
7.2.2	二叉树的性质	145
7.2.3	二叉树的存储结构	147
7.2.4	二叉树的遍历	150
7.2.5	二叉树遍历的应用	153
7.3	线索二叉树	155
7.4	树、森林和二叉树的关系	160
7.4.1	树的存储结构	160
7.4.2	森林与二叉树的转换	163
7.4.3	树和森林的遍历	166
7.5	哈夫曼树及其应用	168
7.5.1	哈夫曼树的定义	168

7.5.2	哈夫曼树的构造	169
7.5.3	哈夫曼树在编码问题中的应用	169
7.6	小结	174
	习题	174
第8章	图	179
8.1	图的定义及基本概念	179
8.2	图的存储结构	183
8.2.1	邻接矩阵表示法	183
8.2.2	邻接表	186
8.2.3	十字链表	190
8.2.4	邻接多重表	191
8.3	图的遍历	193
8.3.1	深度优先搜索	193
8.3.2	广度优先搜索	195
8.4	生成树	197
8.4.1	普里姆算法	198
8.4.2	克鲁斯卡尔算法	201
8.5	最短路径	204
8.5.1	单源最短路径	204
8.5.2	求每一对顶点之间的最短路径	207
8.6	拓扑排序	208
8.7	关键路径	213
8.8	小结	220
	习题	220
第9章	查找	224
9.1	查找的基本概念	224
9.2	线性表的查找	225
9.2.1	顺序查找	225
9.2.2	折半查找	227
9.2.3	分块查找	229
9.3	树表的查找	232
9.3.1	二叉查找树	232
9.3.2	平衡二叉树	238
9.4	哈希表的查找	243
9.4.1	哈希表	243
9.4.2	构造哈希表的基本方法	244
9.4.3	解决冲突的方法	246

9.5	各种查找方法的比较	248
9.6	小结	249
	习题	250
第10章	排序	252
10.1	排序的基本概念	252
10.2	内部排序	254
10.2.1	插入排序	254
10.2.2	冒泡排序	258
10.2.3	快速排序	259
10.2.4	选择排序	262
10.2.5	归并排序	269
10.2.6	基数排序	271
10.3	内部排序方法比较	276
10.4	外部排序简介	277
10.5	小结	277
	习题	278
第11章	文件	280
11.1	外存储器的介绍	280
11.2	有关文件的概念	281
11.2.1	文件及其类别	282
11.2.2	文件的操作	283
11.3	文件的组织	285
11.3.1	顺序文件	285
11.3.2	索引文件	286
11.3.3	散列文件	292
11.3.4	多关键字文件	294
11.4	小结	296
	习题	296
参考文献	298

第1章

绪论

在深入学习数据结构之前,首先了解学习数据结构的意义、什么是数据结构及数据结构的一些相关概念等内容,这对于深刻理解书中后面章节的内容会有很大的帮助。

1.1 数据结构的重要性

在计算机发展的初期,人们使用计算机主要是处理数值的计算问题,程序设计人员也主要把精力集中在程序设计的技巧上。随着计算机应用领域的扩大和硬件的发展,计算机对信息的处理加工已从单一的数值计算发展到大量地解决非数值问题,其加工处理的信息也由简单的数值发展到字符、图像、声音等具有复杂结构的数据。数据结构这门学科随着计算机数据的复杂化而产生并发展起来了。

非数值问题的数据之间的相互关系一般无法完全用数学方程式加以描述,并且数据的表示方法和组织形式直接关系到程序对数据的处理效率,而系统程序和许多应用程序的规模很大,结构复杂,这时人们考虑问题的关键已不再是分析数学和计算方法,而是放在是否能设计出合适的数据结构,有效地解决问题上。

计算机科学是一门研究用计算机进行信息表示和处理的科学。这里面涉及到两个问题:信息的表示和信息的处理。而信息的表示和组成又直接关系到处理信息的程序的效率。随着计算机的普及,信息量的增加,信息范围的拓宽,使许多系统程序和应用程序的规模很大,结构又相当复杂。这就要求人们对计算机程序加工的对象进行系统的研究,即研究数据的特性以及数据之间存在的关系,而数据结构正是进行这种研究的一门课程。

数据结构是计算机专业的核心课程之一,在众多的计算机系统软件和应用软件中都要用到各种数据结构。可以这样说,数据结构不仅是一般程序设计的基础,而且是实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的基础。因此,仅掌握几种计算机语言难以应付众多复杂的研究课题,要想有效地使用计算机,还必须学习数据结构的知识。

瑞士计算机科学家 N. Wirth 教授曾提出这样一个等式:算法+数据结构=程序,这个等式形象地描述了算法、数据结构和程序之间的关系,这里的数据结构指的是数据的逻辑结构和存储结构,而算法就是对数据运算的描述。由此可见,程序设计的实质就是对实

际问题选取一种优秀的数据结构,加之设计一个优秀的算法,而且优秀的算法很大程度上取决于描述实际问题的数据结构。

1.2 基本术语

为了更好地理解数据结构这个概念,首先解释数据结构中的一些常用名词和术语。

数据(data) 是信息的载体,它是描述客观事物的数、字符以及所有能输入到计算机中被计算机程序识别、加工处理的信息的集合。数据不只是通常意义下的整数和实数,随着计算机的广泛应用,数据的范畴也随之拓宽,计算机可以处理的字符串、图像、声音等都可以被称为数据。所以不能只是泛泛地理解数据这个概念。下面进一步解释一下数据的定义。如表 1-1 所示,例如张风的英语成绩为 92 分,92 就是该同学的成绩数据。

表 1-1 学生成绩表

学号	姓名	语文(分)	数学(分)	英语(分)
S01012	张风	85	69	92
S01022	李强	87	73	74
S02013	王海	92	64	84

数据元素(data element) 是数据的基本单位,是对一个客观实体的数据描述。一个数据元素可以由一个或若干个数据项组成。数据元素也被称为结点或记录。

数据项(data item) 是数据的具有独立意义的不可分的最小单位,它是对数据的数据元素属性的描述。数据项也被称为字段、域。

利用上面表 1-1 例子,来说明一下数据项和数据元素,整个表记录的是学生的成绩数据,一个学生的学号、姓名和语文、数学、英语成绩就是其中的一个数据元素,见图 1-1。

学号	姓名	语文(分)	数学(分)	英语(分)
S01012	张风	85	69	92
S01022	李强	87	73	74
S02013	王海	92	64	84

图 1-1 数据元素和数据项

数据对象(data object) 具有相同性质的数据元素的集合就是一个数据对象,它是数据的一个子集。如上例所示,一个班级的成绩表可以看作一个数据对象。例如,集合 $\{1,2,3,4,5,\dots\}$ 是自然数的数据对象,而集合 $\{‘a’,‘b’,\dots,‘z’\}$ 是英文字母表的数据对象。可以看出,数据对象可以是无限的,也可以是有限的。

数据类型(data type) 是具有相同性质的计算机数据的集合及定义在这个数据集合上的一组操作的总称。例如:在 C 语言中的整数类型是集合 $C=\{0,\pm 1,\pm 2,\pm 3,\pm 4,\dots\}$ 及定义在该集合上的加、减、乘、整除和取余等一组操作。数据类型封装了数据存储与操作的具体细节。

每个数据项属于某个确定的基本数据类型,数据类型的种类分原子类型和结构类型。

(1) 原子类型 如果一个数据元素由一个数据项组成,这个数据元素的类型就是这个数据项的数据类型,值在逻辑上不可分解(如 `int i, float j`)。

(2) 结构类型 如果由多个不同的类型的数据项组成,则这个数据元素的类型就是由各数据项类型构造而成的构造类型,值由若干成分按某种结构组成(如 `struct stu`)。上面提到的学生成绩表中,数据项“姓名”的数据类型为字符型,而成绩的数据类型是数值型,所以这个数据元素是一个构造类型。上述成绩表数据用 C 语言的结构体数组 `Class1stu[50]` 来存储。

```
struct Stu
{ /* 数据项 */
int stuID;
char name[20];
int maths_score;
int chinese_score;
int english_score;
} Class1stu[50];
```

不同的高级语言提供的基本数据类型有所不同,在 C 语言中,提供了实型、整型、字符型和指针型等基本数据类型。

抽象数据类型(abstract data type) 就是对象的数学模型,简称为 ADT,是用户在数据类型基础上新定义的数据类型。抽象数据类型定义包括数据组成和对数据的处理操作。可见抽象数据类型是数据和数据使用者的一个接口。

抽象数据类型的定义仅取决于它的一组逻辑特性,而与它在计算机中如何表示和实现无关,只要其数学特性不变,都不影响其外部的使用。从抽象数据类型的角度入手设计软件系统可大大提高软件构件的复用率。抽象数据类型的定义包括数据对象定义、数据关系定义和基本操作定义三部分,其书写格式为:

ADT: 抽象数据类型名

- (1) 数据对象: 数据对象的定义;
- (2) 数据关系: 数据关系的定义;
- (3) 基本操作: 基本操作的定义。

数据结构(data structure) 前面解释了数据结构中的一些重要术语,现在把数据结构这个概念进行一下归纳,数据结构就是数据之间的相互关系(即数据的组织形式)及在这些数据上定义的数据运算方法的集合。

1.3 数据结构的概念

简单地讲,数据结构指的是数据之间的相互关系,即数据的组织形式。它一般包括以下三个方面的内容:

(1) 数据之间的逻辑关系,也称为数据的逻辑结构。

(2) 数据元素及其关系在计算机存储器内的表示,称为数据的存储结构,也就是物理结构。

(3) 数据的运算,即对数据进行的操作。

为了进一步理解数据结构,不妨举一个简单的例子来说明。

如现在有一个学生的基本情况表,排列次序没有任何规律,如表 1-2 所示。表中记录了某校全体学生的姓名和相应的基本信息,现在要求设计一个算法,当给定任何一个学生的姓名时,计算机能够查出该学生的基本信息,如果根本就不存在这个学生,计算机就输出“无此学生记录!”。

这个例子实现的是查找的功能。可以看出,这个算法的设计完全依赖于基本情况表中学生姓名和相应信息在计算机内的存储方式。

表 1-2 学生基本情况表

编号	姓名	年级	年龄(岁)	性别
01	张风	1 年级	12	男
02	李强	1 年级	14	男
03	林海	2 年级	14	男
04	李南	2 年级	10	男
05	韩凤	3 年级	11	女
06	赵加	1 年级	12	女
...

如果学生基本情况表中的学生的姓名是随意排列的,那么,在给定一个学生姓名时,则只能对学生的基本情况表从头到尾逐个与给定的姓名比较,顺序查找直至找到所给的姓名为止,很有可能查找完全部基本情况表还没有找到这个人。虽然这种方法很简单,是线性查询,但是这就会浪费很多时间,效率低。

现在如果将基本情况表进行适当的组织,按字母顺序排列学生的姓名和相应的情况如表 1-3 所示,并且再构造一个字母索引表(表 1-4),这个表用来登记以某个字母开头的第一个学生姓名在基本情况表中的起始位置。当查找某学生的情况时,先从索引表中查到以该字母开头的第一个学生姓名在基本情况表中的起始位置,然后,从此起始处开始查找,而不必去查看以其他字母开头的学生的记录,通过建立这样一种数据组织形式,查找的效率就会有很大的提高。另外,还可以按年级进行排序,然后建立一个年级的索引表,当查询某个年级的学生时,可以先找到这个年级所在的开始位置,然后再查询。这样就大大提高了查找速度。

表 1-3 按开头字母排序的基本情况表

编号	姓名	年级	年龄(岁)	性别
...
11	韩凤	3 年级	11	女
...
24	李强	1 年级	14	男
25	李南	2 年级	10	男
26	林海	2 年级	14	男
...
87	张凤	1 年级	12	男
88	赵加	1 年级	12	女
...

表 1-4 字母索引表

开头字母	编号	开头字母	编号
...
H	11	Z	87
...
L	24		

对于不同的存储结构,就要构造出完全不同的算法。算法和数据结构是密切相关的,算法依赖于具体的数据结构,数据结构直接关系到算法的选择和效率。

此外,当新的学生入校时,学生基本情况表就需要添加新生的姓名和相关的信息;在学生毕业或转学时,应从基本信息表中删除他的记录。这就要求在已安排好的结构上进行插入和删除操作。除此之外,还可能对学生基本情况表进行修改等运算。对于这些运算,由计算机来完成,就需要设计相应的算法,也就是说,数据结构还需要给出每一种结构的各种运算和算法。

通过上面的讨论,可以直观认为:数据结构是研究数据元素之间的相互关系和这种关系在计算机中的存储表示,并对这种结构定义相应的运算,设计出相应的算法,而且确保经过这些运算后所得到的结果仍然是原来的结构类型。

数据结构讨论的问题主要有:如何以最节省存储空间的方式来表示数据和各种不同的数据结构表示方法及其相关算法;如何有效地改进算法效率使程序的执行速度更快;数据处理的各种技巧,如排序、查找等算法的介绍等。

数据结构包括逻辑结构和存储结构(物理结构)。逻辑结构是在逻辑关系上描述数据的,它与数据在计算机内的存储方式没有关系,可看作是从具体问题中抽象出来的数据模型。其关系表示如表 1-5 所示。