

北京市教育委员会共建项目专项资助
高等学校计算机科学与技术专业核心课程系列规划教材

数据结构与算法

(C语言版)

DATA STRUCTURES AND ALGORITHMS IN C

陈明 编著



教材资源网址:

<http://www.edusources.net>

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

员会共建项目专项资助
高等学校计算机科学与技术专业核心课程系列规划教材

数据结构与算法

(C语言版)

陈明 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书为高等院校计算机及相关专业“数据结构”课程的教学用书，系统地介绍了各种典型的数据结构，内容包括：数据结构概论、线性表、栈与队列、串、数组、树、图、查找、排序、递归、文件等；为了加强对算法的理解，还介绍了算法分析方面的内容。

本书语言与选材精练、概念清晰、注重实用、逻辑性强，对于各章节中所涉及的数据结构与算法都给出了C语言描述，并都附有大量的习题，便于学生理解与掌握。

本书适合作为高等院校计算机及相关专业的教材，也可作为计算机应用技术人员的参考书。

图书在版编目(CIP)数据

数据结构与算法：C语言版/陈明编著. —北京：
中国铁道出版社，2012.1
北京市教育委员会共建项目专项资助 高等学校计算
机科学与技术专业核心课程系列规划教材
ISBN 978-7-113-13665-9

I. ①数… II. ①陈… III. ①数据结构—高等学校—
教材②算法分析—高等学校—教材③C语言—程序设计—
高等学校—教材 IV. ①TP311.12②TP301.6③TP312
中国版本图书馆CIP数据核字(2011)第229208号

书 名：数据结构与算法（C语言版）
作 者：陈 明 编著

策 划：吴宏伟 秦绪好 读者热线：400-668-0820
责任编辑：孟 欣
编辑助理：王 惠
封面设计：付 巍
封面制作：白 雪
责任印制：李 佳

出版发行：中国铁道出版社（100054，北京市西城区右安门西街8号）
网 址：<http://www.edusources.net>
印 刷：三河市华业印装厂
版 次：2012年1月第1版 2012年1月第1次印刷
开 本：787mm×1092mm 1/16 印张：17.5 字数：421千
印 数：1~3 000册
书 号：ISBN 978-7-113-13665-9
定 价：32.00元

版权所有 侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社教材图书营销部联系调换。电话：(010) 63550836

打击盗版举报电话：(010) 63549504

“数据结构”课程是从20世纪70年代开始设立的计算机科学与技术专业的一门专业基础课程，现已成为必修的、重要的核心基础课程。数据是用来说明人类活动的事实观念或事物的一些文字、数字或符号。常用的数据类型分为数值数据和非数值数据两大类：数值数据包括整数、定点数、浮点数等；非数值数据主要有逻辑数据、内码和交换码等。数据的级别由低向高依次为位、字节、字、数据项、数据字段、记录、文件、数据库等。

信息是指对某一特定的目的而言，具有意义的事实与知识，使源数据经系统处理成为决策或参考的依据。数据只是事实的记录，没有特定的目的，而信息则是针对某一问题来收集数据并进行处理，作为决策和参考的依据。通过数据处理可归纳出有价值的信息。常用的数据处理方式有编辑、排序、归并、分配、建档、更新、计算、查找、查询等。

计算机科学是算法和算法变换的科学。数据结构主要是研究数据元素之间的关联方式，通常分为逻辑结构和物理结构两大类，同一逻辑结构可以对应不同的物理结构。程序存储是冯·诺依曼机的重要特征之一，构建计算机系统、利用计算机解决问题都是通过程序来实现。算法是求解问题的计算步骤的描述，是程序的核心和灵魂。算法的设计取决于数据的逻辑结构，而算法的实现依赖于指定的存储结构。在程序设计中，要从数据结构和算法两个方面考虑，才能得到高效而准确的结果。

在非数值计算中，处理对象已从简单数值发展到具有结构的数据，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”课程是人们在程序设计方面的经验总结，学会基本的程序设计，只能解决程序设计中30%的问题，而学会数据结构，则能解决程序设计中80%的问题。

“数据结构”课程是计算机程序设计的重要理论技术基础。通过“数据结构”课程的学习，不仅可以掌握数据结构的基本内容、典型算法和使用方法，还能应用数据结构和算法进行具体应用问题的程序设计，进而提高程序设计能力。

本书介绍最常用的典型数据结构、各种数据结构的逻辑关系、在计算机中的存储表示，以及在数据结构中的运算等。数据的逻辑结构主要包括集合、线性表、树、图等4种基本结构，利用它们可以构成任何复杂的逻辑结构。数据的存储结构主要分为顺序、链接、索引和散列等基本结构，利用它们可以构成各种复杂的存储结构。对于同样的数据，采用不同的逻辑结构和存储结构，对某一运算采用的方法不同，将得到不同的算法，进而在计算机上得到不同的运行空间和存储空间效率。

全书在结构上呈积木式，适于选择性学习；选材注重实践应用，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，典型算法的介绍深入浅出、简洁明了。每章都设有小结和习题，通过这些习题的练习，不仅能加深学生对基本概念和定义的理解，而且通过上机练习，能够提高学生的编程能力和程序调试能力。

由于编者水平有限，书中不足之处在所难免，敬请广大读者批评指正。



2011年10月

第 1 章 数据结构概论	1
1.1 问题的提出	2
1.2 基本概念与术语	3
1.3 数据结构的概念	5
1.4 数据的逻辑结构、存储结构及运算	7
1.4.1 数据的逻辑结构	7
1.4.2 数据的存储结构	8
1.4.3 数据的运算	9
1.4.4 逻辑结构、存储结构及运算的关系	10
1.5 算法与算法特性	10
1.5.1 算法及其特性	10
1.5.2 算法的描述方法	11
1.5.3 算法与程序及数据结构	11
1.6 算法性能分析及算法度量	12
1.6.1 算法性能分析	12
1.6.2 算法度量	12
小结	15
习题	15
拓展实验：电话号码的查询	16
第 2 章 线性表	17
2.1 线性表的定义与运算	18
2.1.1 线性表的定义	18
2.1.2 线性表的抽象数据类型	18
2.2 线性表的顺序存储	19
2.2.1 顺序存储	19
2.2.2 顺序表的运算	21
2.3 线性表的链式存储	24
2.3.1 线性链表及运算	24
2.3.2 静态链表及运算	31
2.3.3 循环链表及运算	32
2.3.4 双向链表及运算	34
2.4 线性表的应用	36
2.4.1 约瑟夫问题	37
2.4.2 一元多项式求和问题	38
2.4.3 集合应用问题	41

小结	43
习题	43
拓展实验: 线性表的合并	44
第 3 章 栈与队列	46
3.1 栈	47
3.1.1 栈的定义	47
3.1.2 栈的顺序存储结构	48
3.1.3 栈的链式存储结构	50
3.2 栈的应用	52
3.2.1 子程序的调用和返回问题	52
3.2.2 数制转换问题	52
3.3 队列	53
3.3.1 队列的定义	53
3.3.2 队列的顺序存储结构	54
3.3.3 队列的链式存储结构	60
3.4 队列的应用	64
3.4.1 设备速度不匹配问题	64
3.4.2 舞伴问题	65
小结	66
习题	66
拓展实验: 算术表达式求值	67
第 4 章 串	68
4.1 串的基本概念	69
4.2 串的存储结构	70
4.2.1 串的静态存储结构	70
4.2.2 串的动态存储结构	71
4.3 串的基本运算	73
4.3.1 串的抽象数据类型定义	73
4.3.2 串的基本运算实现	74
4.4 模式匹配	78
4.4.1 BF 算法	78
4.4.2 KMP 算法	80
4.5 串的应用	84
小结	85
习题	85
拓展实验: 设计简单的文本编辑器	86
第 5 章 数组	87
5.1 数组及其基本操作	87
5.1.1 数组的概念	88
5.1.2 抽象数据类型数组的定义	89

5.2	数组的存储结构	90
5.3	数组在矩阵运算中的应用	93
5.3.1	特殊矩阵的压缩存储	93
5.3.2	稀疏矩阵的压缩存储	94
	小结	102
	习题	102
	拓展实验：一元多项式的值计算	103
第 6 章	树	104
6.1	树的概念	105
6.1.1	树的定义	105
6.1.2	树的表示方法	106
6.1.3	树的基本术语	106
6.1.4	树的 ADT 定义	107
6.2	二叉树	107
6.2.1	二叉树的定义及基本结构	108
6.2.2	二叉树的存储结构	109
6.2.3	二叉树的遍历	112
6.3	线索二叉树	115
6.3.1	二叉树的线索化	115
6.3.2	利用线索遍历	116
6.4	树、森林、二叉树之间的关系	120
6.4.1	树的存储结构	121
6.4.2	森林与二叉树的转换	124
6.4.3	树和森林的遍历	127
6.5	哈夫曼算法及其应用	128
6.5.1	哈夫曼树的定义	128
6.5.2	哈夫曼二叉树的构造	129
6.5.3	哈夫曼树在编码问题中的应用	131
	小结	135
	习题	135
	拓展实验：创建二叉树	138
第 7 章	图	139
7.1	图的概念与 ADT 定义	140
7.1.1	图的概念	140
7.1.2	图的抽象数据类型定义	144
7.2	图的存储结构	144
7.2.1	邻接矩阵	145
7.2.2	邻接表	147
7.2.3	十字链表	150
7.2.4	邻接多重表	152

7.3 图的遍历	153
7.3.1 深度优先搜索	153
7.3.2 广度优先搜索	155
7.4 图的应用	157
7.4.1 生成树	157
7.4.2 最短路径	162
7.4.3 拓扑排序	166
7.4.4 关键路径	170
小结	176
习题	176
拓展实验: 图的深度优先搜索	179
第8章 查找	180
8.1 查找的基本概念	181
8.2 静态查找问题	182
8.2.1 顺序查找	182
8.2.2 二分查找	182
8.3 线性表的查找方法	184
8.3.1 线性查找	184
8.3.2 折半查找	185
8.3.3 分块查找	188
8.4 树表的查找方法	190
8.4.1 二叉查找树	190
8.4.2 平衡二叉树	196
8.4.3 B-树	202
8.5 哈希表的查找方法	203
8.5.1 哈希表	203
8.5.2 构造哈希表的基本方法	205
8.5.3 解决冲突的方法	206
8.5.4 哈希表的查找方法	209
8.6 各种查找方法的比较	210
小结	210
习题	210
拓展实验: 折半查找	212
第9章 排序	213
9.1 排序的基本概念	214
9.2 内部排序	216
9.2.1 插入排序	216
9.2.2 冒泡排序	220
9.2.3 快速排序	221
9.2.4 选择排序	223

9.2.5 归并排序	229
9.2.6 基数排序	231
9.3 内部排序方法比较	234
9.4 内部排序方法的选择	235
9.5 外部排序简介	236
小结	236
习题	236
拓展实验：希尔排序	238
第 10 章 递归	239
10.1 递归的定义与类型	240
10.1.1 递归的定义	240
10.1.2 递归的类型	240
10.2 递归应用举例	240
10.2.1 汉诺塔问题	240
10.2.2 八皇后问题	243
10.3 递归的实现	244
10.4 递归到非递归的转换过程	247
10.5 递归的时间和空间复杂度	250
小结	251
习题	251
拓展实验：汉诺塔问题研究	252
第 11 章 文件	253
11.1 外存储器简介	254
11.2 有关文件的概念	255
11.2.1 文件及其类别	255
11.2.2 文件的操作	256
11.3 文件的组织	258
11.3.1 顺序文件	258
11.3.2 索引文件	259
11.3.3 散列文件	264
11.3.4 多关键字文件	265
小结	267
习题	267
拓展实验：索引文件	268
参考文献	269

第

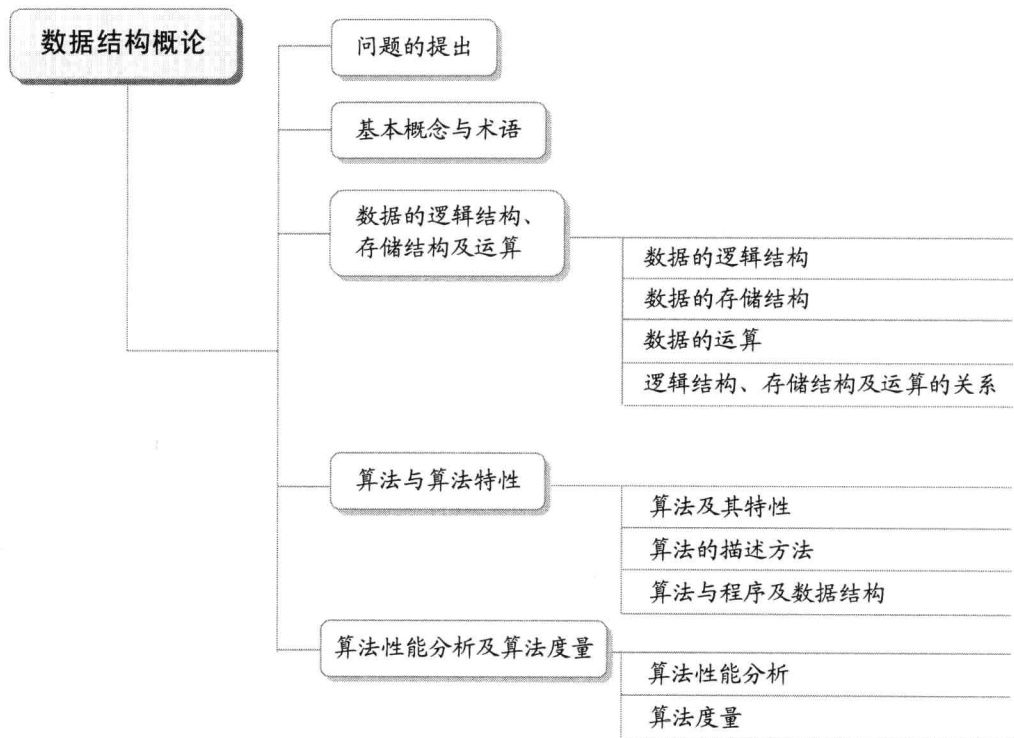
1

章

数据结构概论



本章知识结构图



学习目标

- 了解数据结构的重要性;
- 了解数据结构的基本概念;
- 理解数据的逻辑结构与存储结构;
- 理解数据的运算。

计算机是数据处理机,研究数据在计算机中的组织和处理是计算机科学中的重要课题。数据结构课程就是研究计算机所处理的数据元素间的结构关系及其操作实现的算法。

在深入学习数据结构之前,首先了解一下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基本概念等。这对于深刻理解后面章节的内容将会有很大的帮助。

1.1 问题的提出

在计算机发展的初期,计算机主要用于数值计算,由于计算机的存储容量及计算速度的限制,程序设计人员把主要精力集中在程序设计技巧的研究上。随着计算机应用领域的扩大和软、硬件技术的发展,计算机对信息的处理已从单一的数值计算扩展到解决非数值运算问题,能够处理的信息也由简单的数值扩展到字符、图像、声音等具有复杂结构的数据。而数据结构就是随着计算机的产生和发展而出现的一门较新的计算机课程。

非数值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。目前,系统程序和许多应用程序对数据的运算处理规模很大,而且数据的结构也非常复杂。因此,处理这些问题的关键已不再只是考虑数学分析和计算方法,而是需要设计出合理的数据结构,这样才能有效地解决问题。

计算机科学是一门研究如何用计算机进行信息表示和处理的科学,涉及信息表示和信息处理两个问题。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及效果。随着计算机相关技术应用的扩展,信息量逐渐增加,信息范围也逐渐拓宽,使许多系统程序和应用程序的规模变大,结构变得相当复杂。这就要求计算机程序对待处理的数据对象进行系统的研究,即研究数据的特性以及数据之间的关系,而数据结构正是描述数据的特性以及数据之间关系的一门课程。

【例 1-1】电话号码查询问题。

编写一个查询某个城市或单位的私人电话号码的程序。要求对任意给出的一个姓名,若该人有电话号码,则迅速找到其电话号码;否则,指出该人没有电话号码。

解:要解此问题,首先构造一张电话号码登记表。表中每个结点存放两个数据项:姓名和电话号码。

要想迅速地查找到某人的电话号码,必须根据表中信息的结构和特点,确定存储方式并写出好的查找算法。最简单的方式是将表中的结点按顺序存储在计算机中,查找时从头开始依次查对姓名,直到找出正确的姓名或找遍整个表均没有找到为止。这种查找算法对于一个工作人员较少的单位或许是可行的,但对一个有成千上万私人电话的城市,顺序查找就不实用了。对此,可以选择更有效的索引表来存储电话号码登记表。首先将这张表按姓氏排列,然后另外建立一张姓氏索引表,采用图 1-1 所示的存储结构。查找过程是先在索引表中查对姓氏,然后根据索引表中的地址在电话号码登记表中查对姓名,这样查找登记表时就无须查找其他姓氏的名字了。因此,在这种新的结构上产生的查找算法更为有效。

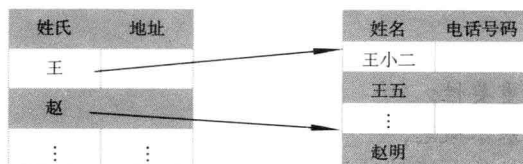


图 1-1 具有索引表的存储结构

数据结构这门课程是前人在程序设计方面积累的程序设计经验，通常来说，学会程序设计基础，可以解决程序设计中 30% 的问题，但学会数据结构，却能解决程序设计中 80% 的问题。数据结构作为计算机专业的核心课程之一，其内容在众多的计算机系统软件和应用软件中都要用到。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是难以应付众多复杂的研究课题的，要想更有效地使用计算机来处理实际问题，就必须打好数据结构的基础。

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，这个等式描述了算法、数据结构和程序之间的关系。数据结构指的是数据的逻辑结构和存储结构，而算法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合理的数据结构，加之设计一些应用在其上的高效的算法，而且算法的好坏很大程度上取决于描述实际问题的数据结构。

1.2 基本概念与术语

为了更好地理解数据结构的概念，先对数据结构中的一些常用名词和术语给出解释。

1. 数据

数据是信息的载体，是对客观事物的描述，这种描述形式可以是数、字符以及所有能输入到计算机中被计算机程序识别、处理的信息的集合。数据不只是数学领域中的整数和实数。在计算机领域中，数据是对客观事物的进一步抽象，数据所能表述的范畴是计算机可以处理的字符串、图像、声音等多种信息。如表 1-1 所示，张风的英语成绩为 92 分，92 就是该学生的成绩数据。

表 1-1 学生成绩表

学 号	姓 名	语 文	数 学	英 语
S01012	张风	85	69	92
S01022	李强	87	73	74
S02013	王海	92	64	84

2. 数据项

数据项是具有独立意义的不可分的最小数据单位，它是对数据结构中的数据元素属性的抽象描述。数据项也被称为字段、域，如图 1-2 所示。

学 号	姓 名	语 文	数 学	英 语
S01012	张风	85	69	92
S01022	李强	87	73	74
S02013	王海	92	64	84

图 1-2 数据元素和数据项

3. 数据元素

数据元素是数据的基本单位，是对一个客观实体数据形式的抽象描述。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。

利用表 1-1 来说明一下数据项和数据元素。整个表描述的是学生的成绩数据，一个学生的某

一门成绩就是其中的一个数据项。一个学生的成绩是一个数据元素，那么这个数据元素由学号、姓名、语文成绩、数学成绩、英语成绩 5 个数据项组成。

4. 数据对象

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做一个数据对象。那么，集合 $\{1,2,3,4,5,\dots\}$ 是自然数的数据对象，而集合 $\{'a', 'b', 'c', 'd', \dots, 'z'\}$ 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。

5. 数据类型

数据类型是将客观事物集抽象描述成具有相同性质的数据元素的集合，以及对这个数据集中数据元素的一组操作的整体表述。例如，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，数据类型分为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结点的数据类型，在逻辑上不可分解。

(2) 结构类型

如果由多个不同类型的数据项组成，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。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的，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 `StuofClass1[50]` 来存储，其形式如下：

```
struct Stu
{ /*数据项*/
    int stuID;
    char name[20];
    int maths_score;
    int chinese_score;
    int english_score;
};
Stu StuofClass1[50]; //StuofClass1[50]表示数据结构体的引用，最大学生数是 50 个
```

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

6. 抽象数据类型

抽象数据类型 (Abstract Data Type, ADT) 是抽象的数学模型，是用户在数据类型基础上定义的新数据类型。抽象数据类型包含数据组成结构的定义以及对建立在其上的处理操作的描述。抽象数据类型是数据和数据使用者的一个接口。调用者通过 ADT 中定义的接口来创建、销毁及控制 ADT 中的数据对象以期满足调用者的需要。

抽象数据类型主要表述建立在抽象层次相对较高的数据相关的逻辑特性，而不关心其在不同的计算机程序语言环境中的表示和实现，只要维持数据的抽象数学特性，数据的使用方式也将保持不变。基于抽象数据类型设计软件系统可以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 3 部分，其格式为：

ADT:抽象数据类型名

- ① 数据对象：对元数据对象结构的定义。
- ② 数据关系：对数据对象之间关系的定义。
- ③ 基本操作：建立在数据对象及数据关系上的相关基本操作的定义。

基本操作主要包括操作名、参数表、初始条件和操作结果 4 部分。其格式为：

操作名(参数表)

操作结果:操作结果描述

在基本操作中，有赋值参数和引用参数。其中，赋值参数只为操作提供输入值，引用参数不仅可以提供输入值，还可以返回操作结果，以&开头，如&T、&G等。

抽象数据类型可以看做描述问题的模型，它独立于具体实现。它的优点是将数据和操作封装在一起，使得用户程序只能通过在 ADT 里定义的某些操作来访问或操作其中的数据，从而实现了信息隐藏。ADT 和类的概念实际上反映了程序或软件设计的两层抽象：ADT 相当于在概念层（或称为抽象层）上描述问题。

由于 C 语言仅有限地支持抽象数据类型概念，为了能从基于抽象数据类型的角度进行介绍，这里对 C 语言进行了适当的补充。补充的主要内容如下：

利用#define 关键字来定义常量和类型；利用 typedef 关键字来描述存储结构；数据元素类型用 ElementType 表示，由用户自行定义。基本操作的描述格式如下：

函数类型 函数名(函数参数表)

```
{
    语句序列
}函数名
```

1.3 数据结构的概念

数据结构是指数据之间的相互关系（即数据的组织形式）及建立在这些数据上的运算方法的集合。为了进一步理解数据结构，举一个简单的例子来说明。

表 1-2 所示的学生基本情况表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，如果不存在这个学生，则计算机输出“无此学生记录！”信息。

表 1-2 学生基本情况表

编 号	姓 名	年 级	年 龄	性 别
01	张风	1 年级	7	男
02	李强	1 年级	8	男
03	林海	2 年级	9	男
04	李南	2 年级	8	男
05	韩凤	3 年级	10	女
06	赵加	1 年级	7	女
⋮	⋮	⋮	⋮	⋮

这个例子实现的是查找功能。查找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。

如果学生基本情况表中的学生姓名是随意排列的,排列次序没有任何规律(见表1-2)。那么,在给定一个学生姓名时,则只能从头到尾对学生基本情况表的各条记录与给定的姓名进行比较,顺序查找直至找到所给定的姓名为止,很有可能表中根本不存在这个人。虽然这种方法很简单,是线性查询,但是浪费时间,效率低下。

如果按首字母顺序排列学生姓名和相应的情况(见表1-3),并且构造一个字母索引表(见表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记录。通过建立这样一种数据组织形式,查找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,就可大大提高查找速度。

表 1-3 按首字母排序的基本情况表

编 号	姓 名	年 级	年 龄	性 别
⋮	⋮	⋮	⋮	⋮
11	韩风	3 年级	10	女
⋮	⋮	⋮	⋮	⋮
24	李强	1 年级	8	男
25	李南	2 年级	8	男
26	林海	2 年级	9	男
⋮	⋮	⋮	⋮	⋮
87	张风	1 年级	7	男
88	赵加	1 年级	7	女
⋮	⋮	⋮	⋮	⋮

选用不同的数据存储结构,建立在其上的算法、操作也不尽相同。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,当新生入学时,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,因此需要设计相应的算法。也就是说,每种数据结构都需要给出各自结构对应的算法。

通过上述讨论,可以认为: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数据之间仍然保持原有的相互关系。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;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,例如排序、查找等算法等。

表 1-4 字母索引表

开 头 字 母	编 号
⋮	⋮
H	11
⋮	⋮
L	24
⋮	⋮
Z	87
⋮	⋮

1.4 数据的逻辑结构、存储结构及运算

1.4.1 数据的逻辑结构

数据结构在形式上可定义为一个二元组：

$$\text{Data_Structure}=(D,R)$$

其中， D 是数据元素的有限集合， R 是 D 上关系的有限集合。

由此可以看出，数据结构由两部分构成：

- ① 数据元素的集合 D ；
- ② 数据元素之间关系的集合 R 。

假设要设计一个事务管理的程序，用来管理 IT 公司程序研发部门的各项事务。现在需要设计一个数据结构，如果要求每个部门由 1 位部门经理、1~4 名项目经理和 1~8 名软件工程师组成，在小组中，1 位部门经理管理 1~4 名项目经理，每位项目经理带领 1 或 2 名由软件工程师组成的团队，得到一个数据结构：

$$\text{Group}=(P,R)$$

其中， P 表示数据元素，包括部门经理、项目经理、软件工程师，即 $P=\{\text{DM},\text{PM}_1,\dots,\text{PM}_n,\text{SE}_{11},\text{SE}_{12},\dots,\text{SE}_{nm}\}$ ， $1\leq n\leq 4$ ， $1\leq m\leq 2$ 。

R 表示小组成员的关系，其关系有两种，部门经理和项目经理： $R_1=\{\langle \text{DM},\text{PM}_i \rangle \mid 1\leq i\leq n, 1\leq n\leq 4\}$ ；项目经理和软件工程师： $R_2=\{\langle \text{PM}_i,\text{SE}_{ij} \rangle \mid 1\leq i\leq n, 1\leq j\leq m, 1\leq n\leq 4, 1\leq m\leq 2\}$ 。

再举一个例子，一周 7 天的数据结构可表示为：

$$\text{Group}=(D,S)$$

$$D=\{\text{星期一},\text{星期二},\text{星期三},\text{星期四},\text{星期五},\text{星期六},\text{星期天}\}$$

$$S=\{\langle \text{星期天},\text{星期一} \rangle,\langle \text{星期一},\text{星期二} \rangle,\langle \text{星期二},\text{星期三} \rangle,\langle \text{星期三},\text{星期四} \rangle,\langle \text{星期四},\text{星期五} \rangle,\langle \text{星期五},\text{星期六} \rangle,\langle \text{星期六},\text{星期天} \rangle\}$$

以上数据结构可用图 1-3 表示。

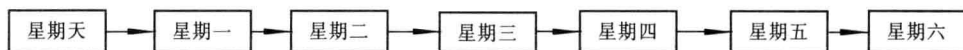


图 1-3 一周 7 天数据结构图示

总而言之，数据结构是相互之间存在一种或多种特定关系的数据元素的集合，这个关系特指数据元素之间的逻辑关系，也称为数据的逻辑结构，它与在计算机上的具体数据存储无关。因此，数据的逻辑结构可以看做从具体的问题中抽象出来的数学模型。

数据的逻辑结构分为 3 种典型结构：集合、线性结构、非线性结构。

① 集合中元素间为松散的关系，只是同属于一个集合而已，如各种颜色属于色彩集合。

② 线性结构的逻辑特征是有且仅有一个起始结点和一个终端结点，并且所有结点只有一个直接前驱结点和一个直接后继结点，如线性表、队列等，结点之间是一一对一的关系，如前面的学生基本情况表中每个数据元素之间的关系。

③ 非线性结构的特征是一个结点可能有多个直接前驱结点或多个直接后继结点，如树、图等，树的结点之间是一对多的关系，图的结点之间是多对多的关系。这些将在后面的章节中介绍。