

信息与电子学科百本精品教材工程

新编计算机类本科规划教材

数据结构 (Java版)

叶核亚 编著 陈本林 主审

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



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

新编计算机类本科规划教材

数据结构(Java 版)

叶核亚 编著
陈本林 主审

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书全面阐述数据结构方面的基本理论, 主要内容包括线性表、栈、队列、串、数组、广义表、树、二叉树、图等基本的数据结构以及查找和排序算法。

本书用 Java 语言定义和实现数据结构及算法。全书结构清楚, 内容丰富, 章节安排合理。叙述深入浅出, 循序渐进。示例典型实用, 算法严谨规范, 算法和程序全部调试通过。

本书适合作为高等院校计算机及相关专业本、专科学生教材, 也可作为从事计算机软件开发和工程应用人员的参考书。

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 (Java 版) / 叶核亚编著. —北京: 电子工业出版社, 2004.5

新编计算机类本科规划教材

ISBN 7-5053-9857-1

I. 数… II. 叶… III. ①数据结构—高等学校—教材 ②JAVA 语言—程序设计—高等学校—教材 IV. ①

TP311.12 ②TP312

中国版本图书馆 CIP 数据核字 (2004) 第 034213 号

策划编辑: 章海涛

责任编辑: 章海涛

印 刷: 北京大中印刷厂

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

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 15.75 字数: 400 千字

印 次: 2004 年 5 月第 1 次印刷

印 数: 1~5 000 册 定价: 19.50 元

凡购买电子工业出版社的图书, 如有缺损问题, 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。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。

前 言

“数据结构”课程在计算机类专业学生培养中的地位十分重要。在计算机中，现实世界被抽象为数据或数据模型。“数据结构”课程的任务是讨论数据的各种逻辑结构和在计算机中的存储结构，以及各种操作的算法设计。“数据结构”课程的主要目的是培养学生掌握处理数据和编写高效率软件的基本方法，从而为以后进行软件开发和应用及进一步学习后续专业课程打下坚实的基础。“数据结构”课程讨论的知识内容是软件设计的理论基础，数据结构课程介绍的技术方法是软件设计中使用的基本方法。因此，“数据结构”课程是计算机学科本科（专科）的核心课程，是培养程序设计技能的必不可少的重要环节。

本书具有以下特点：

1. 内容全面，叙述准确

本书介绍了线性表、栈、队列、串、数组、广义表、树和二叉树、图等基本的数据结构，阐明了数据模型的逻辑结构，讨论了它们在计算机中的存储结构以及能进行的各种操作和这些操作的具体实现。另外，本书讨论了软件设计中应用频繁的查找和排序问题，给出了多种经典查找和排序算法。

本书以基本数据结构和经典算法为主线，全面、准确地阐述了“数据结构”课程的主要内容。本书概念清楚，用语规范，思路清晰。书中示例选择合适，算法分析透彻，程序结构严谨、规范，体现了良好的程序设计素养。

2. 采用 Java 语言描述数据结构

随着计算机技术日新月异的变化及网络化的发展趋势，Java 语言以其完全面向对象、简单高效、与平台无关、支持多线程、具有安全性和健壮性等特点，已成为目前最具吸引力且发展势头迅猛的程序设计语言。

Java 语言是完全面向对象的，具有诸多传统程序设计语言无法比拟的优点。Java 语言的语法类似 C++ 语言，但除去了 C++ 中那些模糊和容易引起错误的特性，同时引入了很多独特的高级特性。Java 语言比 C++ 更容易学习，且用 Java 语言编写的程序可读性更好。

本书用 Java 语言定义和实现了全部的数据结构。

3. 具有丰富翔实的例题

本书注重理论与实践相结合，注重基本知识的传授与基本技能的培养。本书除例题外，还安排了思考题、课堂练习题、上机实习题以及综合程序设计题等实践环节。这些为巩固读者的理论知识，提高读者的软件设计能力起到了辅助作用，为读者以后进行系统软件和应用软件的开发研究打下了坚实的基础。

书中的所有程序全部调试通过。

本书由叶核亚主编，陈立参编。其中叶核亚编写第 1、2、3、4、6、7、8、9 章，陈立编写第 5 章。本书由陈本林教授主审。陈教授认真细致地审阅了全书，并提出了许多宝贵意见。本书在编写过程中还得到了廖雷、陈瑞、陈建红、阚建飞、李林广、王青云、刁翔等老师的大力帮助，在此一并表示感谢。

尽管作者写作时非常认真和努力，但由于水平有限，时间紧迫，书中难免有疏漏之处，敬请读者批评指正。

为了使用本书的教师和读者的需要，我们将免费提供本书中所有的图、表和程序，请发E-mail 至 unicode@phei.com.cn 索要。

编著者
2004 年 3 月

阅 读 推 介

“数据结构”课程是计算机及其相关专业的一门核心必修课。本教材系统、全面地阐述了数据结构的基本理论，详细介绍了线性表、栈、队列、串、数组、广义表、树、二叉树、图等基本的数据结构以及查找和排序算法。全书内容取材恰当，组织合理，理论叙述准确、清楚，文字流畅。

“数据结构”课程既是一门理论课程，又是一门锻炼程序设计能力的实践课程。本教材既注重基本知识的传授，又注重对读者基本技能的培养。全书精心安排了大量有特色的例题和完整的实例程序，体现出作者丰富的教学经验和理论研究能力。

计算机软件开发方法是不断发展的，“数据结构”课程的内容也应随着软件开发方法的不断发展而发展。目前，面向对象的软件分析和设计技术已发展成为软件开发的主流方法，因此，用面向对象的方法描述数据结构就成为“数据结构”课程改革的必然。

用面向对象的观点描述具体的数据结构问题时，首先涉及选用哪种面向对象的高级语言的问题。

Java 语言是一种完全面向对象的语言。Java 语言具有平台无关性、分布式、可移植性、可重用性、健壮性、安全性等特点，这些优点是诸多传统程序设计语言所无法比拟的。所以 Java 语言能够在众多程序设计语言中脱颖而出，成为当今广泛使用的主流语言之一。

Java 语言继承了 C++ 语言的优秀特性，语法上类似 C++，易于学习。Java 语言比 C++ 语言简洁、精炼，类之间的继承关系简明清晰，程序的可读性更好。

采用 Java 语言既能够准确描述基本数据结构，充分表达算法的设计与实现，又能够进一步以面向对象的软件设计思想锻炼程序设计能力。所以，采用 Java 语言描述数据结构是一种发展趋势。

目前，除部分译著外，由国内作者编写的同类教材并不多见。

本教材采用 Java 语言以面向对象方法设计并实现了全部的数据结构及算法，具有一定的创新性、前瞻性，体现出作者扎实的理论功底和应用设计能力。

本教材适用于高等学校计算机及相关专业的本、专科学生。

2004 年 3 月

目 录

第 1 章 绪论	1
1.1 数据结构的基本概念	1
1.1.1 抽象数据类型与数据结构	1
1.1.2 数据的逻辑结构	3
1.1.3 数据的存储结构	3
1.1.4 数据的操作	5
1.2 算法与算法设计	6
1.2.1 算法	6
1.2.2 算法设计	8
1.2.3 算法分析	9
1.3 Java 语言简介	11
1.3.1 Java 的安装、编辑、编译和运行	11
1.3.2 数据类型与流程控制	13
1.3.3 类与对象	17
1.3.4 类的继承性与多态性	19
1.3.5 Java 的接口、内部类与包	21
1.3.6 异常处理	23
1.3.7 Java 的标准数据流	25
习题 1	26
实习 1	26
第 2 章 线性表	27
2.1 线性表的概念	27
2.1.1 线性表的抽象数据类型	27
2.1.2 线性表的顺序存储结构	28
2.1.3 线性表的链式存储结构	36
2.2 线性链表	37
2.2.1 单向链表	37
2.2.2 单向循环链表	46
2.2.3 双向链表	49
2.2.4 双向循环链表	52
2.3 串	53
2.3.1 串的定义	53
2.3.2 串的存储结构	54
2.3.3 串的操作	54
习题 2	62
实习 2	62

第3章 排序	63
3.1 排序的基本概念	63
3.2 插入排序	64
3.2.1 直接插入排序	64
3.2.2 希尔排序	70
3.3 交换排序	73
3.3.1 冒泡排序	73
3.3.2 快速排序	75
3.4 选择排序	79
3.5 归并排序	82
习题3	88
实习3	88
第4章 栈与队列	90
4.1 栈	90
4.1.1 栈的定义	90
4.1.2 栈的抽象数据类型	91
4.1.3 栈的存储结构及实现	92
4.1.4 栈的应用举例	97
4.2 队列	104
4.2.1 队列的定义	104
4.2.2 队列的抽象数据类型	105
4.2.3 队列的存储结构及实现	105
4.2.4 队列的应用举例	111
4.3 递归	113
习题4	118
实习4	118
第5章 数组和广义表	120
5.1 数组	120
5.1.1 一维数组	120
5.1.2 多维数组	121
5.2 稀疏矩阵	124
5.2.1 三元组的顺序存储结构	124
5.2.2 三元组的链式存储结构	128
5.3 广义表	131
5.3.1 广义表的概念	131
5.3.2 广义表的存储结构	133
习题5	135
实习5	135
第6章 树和二叉树	136
6.1 树	136

6.1.1	树的定义	137
6.1.2	树的术语	137
6.1.3	树的广义表形式表示	138
6.2	二叉树的定义及性质	139
6.2.1	二叉树的定义	139
6.2.2	二叉树的性质	140
6.2.3	二叉树的存储结构	141
6.2.4	声明二叉树类	142
6.3	二叉树的遍历	143
6.3.1	二叉树遍历的概念	143
6.3.2	二叉树遍历的递归算法	144
6.3.3	建立二叉树	145
6.3.4	二叉树遍历的非递归算法	155
6.3.5	层次遍历二叉树	157
6.4	线索二叉树	159
6.4.1	线索二叉树的定义	159
6.4.2	中序线索二叉树	161
6.5	堆排序	167
6.6	树与二叉树的转换	171
	习题 6	172
	实习 6	173
第 7 章	查找	174
7.1	查找的基本概念	174
7.2	线性表的查找	176
7.2.1	顺序查找	176
7.2.2	折半查找	178
7.2.3	分块查找	182
7.3	二叉排序树及其查找算法	189
7.4	哈希查找	194
	习题 7	199
	实习 7	199
第 8 章	图	200
8.1	图的基本知识	200
8.1.1	图的定义	200
8.1.2	结点的度	202
8.1.3	子图	202
8.1.4	路径、回路及连通性	203
8.2	图的存储结构	204
8.2.1	邻接矩阵	204
8.2.2	邻接表	205

8.3 图的遍历	206
8.3.1 深度优先遍历	207
8.3.2 广度优先遍历	211
8.4 最小代价生成树	214
8.4.1 树与图	214
8.4.2 生成树	215
8.4.3 最小代价生成树	216
8.5 最短路径	218
习题 8	219
实习 8	220
第 9 章 综合应用设计	221
9.1 用“预见算法”解骑士游历问题	221
9.2 综合应用实习	229
附录 A ASCII 码表	233
附录 B Java 关键字表	234
附录 C Java 部分类库表	236
参考文献	242

第1章 绪 论



教学要点

软件设计是计算机学科各个领域的核心。软件设计时要考虑的首要问题是数据的表示、组织和处理方法,因为数据的表示、组织和处理方法直接关系到软件的工程化程度和软件的运行效率。

随着计算机技术的飞速发展,计算机的应用从早期的科学计算扩大到控制、管理和数据处理等各个领域。计算机处理的对象也从简单的数值数据,发展到多媒体数据。各种软件系统处理的数据量越来越大,数据的类型越来越多,数据的结构越来越复杂。因此,针对实际问题,如何合理地组织数据,如何建立合适的数据结构,如何设计优秀的算法,则是软件系统设计的重要问题,而这些就是“数据结构”这门课程讨论的主要内容。

数据结构设计和算法设计是软件系统设计的核心。在计算机领域流传着一句经典名言,就是“数据结构+算法=程序”(瑞士 Niklaus Wirth 教授)。这句话简洁明了地说明了程序(或软件)和数据结构与算法的关系,也简洁明了地说明数据结构课程的重要性。

作为绪论,本章将讨论后续章节中频繁使用的数据、数据类型、数据结构、算法等基本概念,以及算法分析的基本方法。

建议本章授课4学时。

1.1 数据结构的基本概念

1.1.1 抽象数据类型与数据结构

1. 数据

描述客观事物的数字、字符以及所有能输入到计算机中并能为计算机接受的各种符号集合统称为数据(data)。数据是程序的处理对象。例如,学生成绩管理程序处理的数据是每个学生的情况,包括学号、姓名、年龄、各科成绩等,编译程序处理的数据是用各种高级程序设计语言书写的源程序。近年来,随着技术的进步,数据的形式越来越多,如多媒体技术中涉及的视频和音频信号,经采集转换后都能形成计算机可接受和处理的数据。

表示一个事物的一组数据称为一个数据元素(data element)。数据元素是数据的基本单位,构成数据元素的数据称为该数据元素的数据项(data item)。

2. 数据类型

类型是一组值的集合。数据类型(data type)是指一个类型和定义在该类型上的操作集合。数据类型定义了数据的性质、取值范围以及对数据所能进行的各种操作。例如,Java 中

整数类型 `int` 的值集是 $\{-2^{32}, \dots, -2, -1, 0, 1, 2, \dots, 2^{32}-1\}$ ，还包括对这个整数类型进行的加 (+)、减 (-)、乘 (*)、除 (/) 和求模 (%) 操作。

高级程序设计语言都提供了一些基本数据类型，如 `Java` 中有 `int`、`long`、`float`、`double`、`char`、`String` 等基本数据类型。利用这些基本数据类型，软件设计人员还可以设计出各种复杂的数据类型。例如，学生姓名可以用字符串类型 `String` 表示，年龄可以用整数类型 `int` 表示，成绩可以用浮点类型 `float` 表示等，而包括学号、姓名、成绩等的学生情况就是一种复杂的数据类型。

3. 抽象数据类型

抽象数据类型 (Abstract Data Type, ADT) 是指一个逻辑概念上的类型和该类型上的操作集合。

没有定义具体数据类型的数据元素称为抽象数据元素。

从定义看，数据类型和抽象数据类型的定义基本相同。数据类型和抽象数据类型的不同之处仅仅在于：数据类型指的是高级程序设计语言支持的基本数据类型，而抽象数据类型指的是在基本数据类型支持下用户新设计的数据类型。“数据结构”课程主要讨论表、栈、队列、串、数组、树和二叉树、图等典型的数据结构，这些典型的数据结构就是一个个不同的抽象数据类型。

表 1-1 学生信息表

学号	姓名	性别	年龄
20020001	王红	男	18
20020002	张明	男	19
20020003	吴宁	女	18

例如，要描述学生信息，可包括学生的学号、姓名、性别、年龄等数据。学生的学号、姓名、性别、年龄等数据就构成学生情况描述的数据项；包括学号、姓名、性别、年龄等数据项的一组数据就构成学生信息的一个数据元素。表 1-1 是一个有 3 个数据元素的学生信息表。

本书采用 `Java` 语言描述抽象数据类型，抽象数据元素可以声明为 `Java` 语言中的 `Object` 类。表 1-1 的学生信息表可以声明为如下的类，其中数据元素类型是 `Java` 的基本数据类型。

```
class Student
{
    String number;
    String name;
    String sex;
    int age;
}
```

4. 数据结构

计算机处理的数据不是杂乱无章的，而是有着内在联系的。只有分析清楚它们的内在联系，对大量的、复杂的数据才能进行合理地组织和有效地处理。例如，由一个班级的学生组成的一批数据，按学号排列具有“顺序”关系，按班长、组长、组员排列具有“层次”关系。这种关系不因数据的改变而改变。

对一个数据元素集合来说，如果在数据元素之间存在一种或多种特定的关系，则称为数据结构 (data structure)。因此，“结构”就是指数据元素之间存在的关系。

数据结构与数据类型是两个不同的概念。数据类型研究的是每类数据所共有的特性，关注的是数据集合是怎样的，该数据集合上允许进行的操作有哪些；数据结构研究的是相互关联的

数据之间的关系，数据结构关注数据之间的关系是怎样的。例如，一个班级的学生按学号排列就是顺序关系，按班长、组长、组员排列就具有层次关系；而祖父、父亲、我、儿子、孙子间的辈份关系也是层次关系。

1.1.2 数据的逻辑结构

数据结构课程主要讨论三方面的问题：数据的逻辑结构、数据的存储结构和数据的操作。

数据的逻辑结构是对数据元素之间逻辑关系的描述，它可以用一个数据元素的集合和定义在此集合上的若干关系来表示，数据的逻辑结构经常被简称为数据结构。

按照数据元素之间存在的逻辑关系的不同数学特性，基本的数据结构有 3 种：线性结构，树结构和图结构。

1. 线性结构

线性结构的定义是：除第一个和最后一个数据元素外，每个数据元素只有一个前驱数据元素和一个后继数据元素，第一个数据元素没有前驱数据元素，最后一个数据元素没有后继数据元素。

线性结构如图 1.1 (a) 所示，其中数据元素 B 有一个前驱数据元素 A，有一个后继数据元素 C。

2. 树结构

树结构的一般定义是：除根结点外，每个数据元素只有一个前驱数据元素，可有零个或若干个后继数据元素，根结点没有前驱数据元素。

树结构如图 1.1 (b) 所示，其中数据元素 B 有一个前驱数据元素 A，有两个后继数据元素 D 和 E。

3. 图结构

图结构的一般定义是：每个数据元素可有零个或若干个前驱数据元素，可有零个或若干个后继数据元素。

图结构如图 1.1 (c) 所示，其中数据元素 E 有两个前驱数据元素 B 和 C。

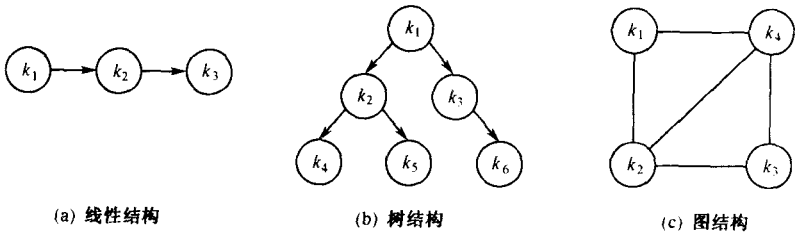


图 1.1 三种基本的数据结构

图 1.1 以图示法表示数据的逻辑结构。图示法中，圆圈表示一个数据元素，圆圈中的字符表示数据元素的标记或值，连线表示数据元素间的关系。

1.1.3 数据的存储结构

任何需要计算机进行管理和处理的数据元素都必须首先按某种方式存储在计算机中。数

据元素在计算机中的存储表示方式称为数据的存储结构，也称为物理结构。数据的存储结构要能正确地表示出数据元素间的逻辑关系。

数据的逻辑结构是从逻辑关系角度观察数据，它与数据的存储无关，是独立于计算机的。而数据的存储结构是逻辑结构在计算机中的实现，它是依赖于计算机的。

1. 两种存储结构

数据存储结构的基本形式有 2 种：顺序存储结构和链式存储结构。

顺序存储结构是把数据元素存储在一块地址连续的空间中，其特点是逻辑上相邻的数据元素在物理上也相邻，数据间的逻辑关系表现在数据元素的存储位置关系上。

例如，采用高级程序设计语言中的数组可以实现顺序存储结构，数组元素之间的顺序体现了线性结构数据元素之间的逻辑次序。

指针是指向物理存储单元地址的变量。由数据元素域和指针域组成的一个整体称为一个结点 (node)。链式存储结构是使用指针把相互直接关联的结点 (即直接前驱结点或直接后继结点) 连接起来，其特点是逻辑上相邻的数据元素在物理上 (即内存存储位置上) 不一定相邻，数据间的逻辑关系表现在结点的链接关系上。

就像自行车的链条是由每一节链串起来的一样，结点在链条中的位置可根据需要重新组织形成新的链条。

对于数据元素是 {A, B, C, D} 的线性结构，其顺序和链式存储结构如图 1.2 所示。

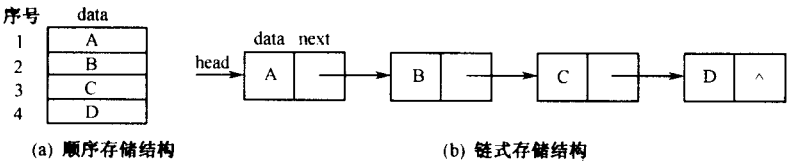


图 1.2 两种存储结构

在顺序存储结构中，所有的存储空间都被数据元素占用了；而链式存储结构中，每个结点除了要保存数据元素外，还要保存指向后继结点的链信息。因此，每个结点至少由两部分组成：数据域——保存数据元素，链——指向后继结点。

【例 1.1】 学生信息表的两种存储结构。

一个班级的学生信息表按学号排列起来，构成一个线性表，其顺序与链式存储结构如图 1.3 所示。

顺序存储结构和链式存储结构是两种最基本、最常用的存储结构。除此之外，利用顺序存储结构和链式存储结构进行组合，还可以有一些更复杂的存储结构。

2. 存储密度

如果所有的存储空间都用来存储数据元素，则这种存储结构是紧凑结构，否则称为非紧凑结构。显然，顺序存储结构是紧凑结构，链式存储结构是非紧凑结构。

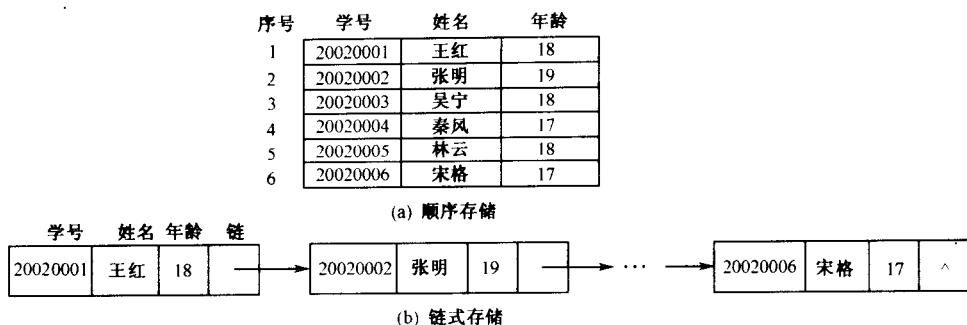


图 1.3 学生信息表的两种存储结构

结构的存储密度定义为数据本身所占的存储量和整个结构所占的存储总量之比，即

$$d = \frac{\text{数据本身所占的存储量}}{\text{整个结构所占的存储总量}}$$

紧凑结构的存储密度为 1，非紧凑结构的存储密度小于 1。存储密度越大，则存储空间的利用率越高。但是存储附加的信息会带来操作上的方便。

3. 选择数据的存储结构

在逻辑结构的基础上，选择一种合适的存储结构，使得在以下两方面的综合性能最佳：数据操作所花费的时间少，占用的存储空间少。

计算机运行任何程序都要花费一定的时间和占用一定的内存空间。例如，存储一个线性表，当不需要频繁插入和删除时，采用顺序存储结构是可行的。因为这时存储密度大，占用的空间少。而当插入和删除操作很频繁时，就需要采用链式存储结构。因为这时虽然占用的空间较多，使存储密度变小，但节省了操作的时间。这是以空间为代价换取了时间，在总的时间和空间性能上仍是可行的。

1.1.4 数据的操作

对一种数据类型的数据进行的某种处理称为数据的操作，对一种数据类型的数据进行的所有操作称为数据的操作集合。

数据的操作是定义在数据的逻辑结构上的，每种逻辑结构都有一个操作集合，不同的逻辑结构有不同的操作。操作的具体实现与存储结构有关。下面列举线性结构、树结构和图结构中几种常用的操作：

- 访问数据元素。
- 统计数据元素个数。
- 更新数据元素值。
- 插入数据元素，在数据结构中增加新的结点。
- 删除数据元素，将指定数据元素从数据结构中删除。
- 查找——在数据结构中查找满足一定条件的数据元素。插入、删除、更新操作都包括一个查找操作，以确定所需插入、删除、更新数据元素的确切位置。
- 排序——在线性结构中数据元素数目不变的情况下，将数据元素按某种指定的顺序重新排列。

【例 1.2】 学生信息表的操作。

对图 1.3 所示的学生信息表进行操作，结果如下：

- 访问学号为“20020003”的数据元素，得到学生姓名为“吴宁”，年龄为 18。
- 统计数据元素个数，共 6 个学生。
- 更新学号为“20020005”、年龄值为 20 的数据元素（如图 1.4（a）所示）。
- 插入学号为“20020007”、姓名为“刘强”、年龄为 18 的数据元素（如图 1.4（a）所示）。
- 删除学号为“20020004”的数据元素（如图 1.4（b）所示）。
- 查找年龄等于 20 的数据元素，序号为 4（如图 1.4（b）所示）。
- 按学号排序。

序号	学号	姓名	年龄
1	20020001	王红	18
2	20020002	张明	19
3	20020003	吴宁	18
4	20020004	秦风	17
5	20020005	林云	20
6	20020006	宋格	17
7	20020007	刘强	18

(a) 更改、插入结点

序号	学号	姓名	年龄
1	20020001	王红	18
2	20020002	张明	19
3	20020003	吴宁	18
4	20020005	林云	20
5	20020006	宋格	17
6	20020007	刘强	18
7			

(b) 删除结点

图 1.4 学生信息表的操作

1.2 算法与算法设计

1.2.1 算法

通俗地说，算法是对问题求解过程的一种描述，是为解决一类问题给出的一个确定的、有限长的操作序列。

1. 算法定义

曾获图灵奖的著名计算科学家 D.Knuth 对算法做过一个为学术界广泛接受的描述性的定义：一个算法（algorithm），就是一个有穷规则的集合，其规则确定了一个解决某一特定类型问题的操作序列。算法的规则必须满足如下 5 个特征：

- 有穷性——算法必须在执行有穷步骤之后结束。
- 确定性——算法的每一个步骤必须是确切定义的。
- 输入——算法有零个或多个输入。
- 输出——算法有一个或多个输出，与输入有某种特定关系。
- 可行性——算法中有待执行的操作必须是相当基本的，即是说，它们原则上都是能够精确地进行的，而且用笔和纸做有穷次就可以完成。

有穷性和可行性是算法最重要的两个特征。

2. 算法的描述

算法可用文字、高级程序设计语言或类似于高级程序设计语言的伪码描述。此时，算法是由语义明确的操作步骤组成的有限序列，它精确地指出怎样从给定的输入信息得到要求的输出信息。

【例 1.3】 学生信息表的顺序查找算法。

在学生信息表中，按“姓名”进行顺序查找（sequential search）的算法思路为：对于给定值 k ，从线性表的一端开始，依次比较每个元素的姓名栏，如果存在姓名与 k 相同的数据元素，则查找成功；否则查找不成功。

这种方法是可行的，但查找范围大，耗时多，查找不成功时，需要将表中的所有元素全部比较后才能确定。

设学生信息表按顺序存储，按“姓名”查找值 “name=“宋格””，顺序查找算法描述如下：

- 设 $i=1$ ，比较第 i 个元素的姓名是否等于 name，如果相等，则查找成功，查找过程结束；否则， $i++$ ，继续比较。
- 学生信息表中，所有元素的姓名都不等于 name，则查找不成功。

学生信息表的顺序查找过程如图 1.5 所示。

序号	学号	姓名	年龄	
$i=$ 1	20020001	王红	18	第1个元素的姓名等于 name?
2	20020002	张明	19	第2个元素的姓名等于 name?
3	20020003	吴宁	18	第3个元素的姓名等于 name?
4	20020005	林云	20	第4个元素的姓名等于 name?
5	20020006	宋格	17	第5个元素的姓名等于 name? 是，则查找成功
6	20020007	刘强	18	
7				

图 1.5 学生信息表的顺序查找

3. 算法与数据结构

算法是建立在数据的逻辑结构与存储结构上的。数据结构和算法之间存在着本质的联系，失去一方，另一方就没有意义。在研究一种数据结构时，总是离不开对这种数据结构需要施加的各种操作，而且只有通过对这些操作的算法的研究，才能更清楚地理解这种数据结构的意义和作用。反过来，在研究一种算法时，也总是自然地联系到该算法建立在什么样的数据结构之上。

(1) 对于同样的逻辑结构和存储结构，根据问题的不同要求，采用不同的算法。

【例 1.4】 字典的分块查找算法。

一部字典是一个顺序存储的线性表，且已按字母顺序排序，与学生成绩表的逻辑结构和存储结构都相同。因学生成绩表的数据量较小，可以采用顺序查找算法；而字典的数据量较大，则必须采用分块查找算法。

例如，在字典中查找单词 state。我们通常先查找索引表，找到以字母 s 开头的页码，再根据 state 后几个字母，就可快速准确地定位 state，查阅 state 的含义。那么，如何知道以字母 s 开头的单词的起始页码？为此，每部字典都设计了一个索引表，指出每个字母对应单词的起始页码（如图 1.6 所示）。

索引表			字典	
序号	首字母	起始页	页码	单词
1	a	1	1	aardvark
2	b	12		abstract
3	c	23		...
...	...	...		
19	s	450	12	baal
...	...	...		...
			461	state

图 1.6 字典与其索引表示意图

一部字典可以看成是由首字母相同、大小不等的若干块（block）组成，每个块的起始页码由索引表给出。

在字典中查找给定单词 word，分块查找（blocking search）算法必须分两步进行：

- 根据 word 的首字母，查找索引表，确定 word 所在块的起始页码。