



普通高等教育规划教材

数据结构(C++版)

叶核亚 主编



 机械工业出版社
CHINA MACHINE PRESS



普通高等教育规划教材

数据结构（C++版）

主 编 叶核亚

副主编 黄 纬

参 编 孙水华

主 审 朱战立



机械工业出版社

本书全面阐述了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 C++ 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。

图书在版编目 (CIP) 数据

数据结构: C++ 版/叶核亚主编. —北京: 机械工业出版社, 2004.8

普通高等教育规划教材

ISBN 7-111-14872-X

I. 数... II. 叶 ... III. ① 数据结构—高等学校—教材② C 语言—程序设计—高等学校—教材 IV TP311.12
② TP312

中国版本图书馆 CIP 数据核字 (2004) 第 066652 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 王小东

封面设计: 饶 薇 责任印制 李 妍

北京机工印刷厂印刷 · 新华书店北京发行所发行

2005 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16 · 16.5 印张 · 404 千字

定价: 23.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、88379646

封面无防伪标均为盗版

普通高等教育应用型人才培养规划教材编审委员会名单

主 任：刘国荣 湖南工程学院

副主任：左健民 南京工程学院

陈力华 上海工程技术大学

鲍 泓 北京联合大学

王文斌 机械工业出版社

委 员：（按姓氏笔画排序）

刘向东 华北航天工业学院

任淑淳 上海应用技术学院

何一鸣 常州工学院

陈文哲 福建工程学院

陈 峻 扬州大学

苏 群 黑龙江工程学院

娄炳林 湖南工程学院

梁景凯 哈尔滨工业大学(威海)

童幸生 江汉大学

计算机科学与技术专业分委员会名单

主 任：黄陈蓉 南京工程学院

副主任：吴伟昶 上海应用技术学院

委 员：（按姓氏笔画排序）

汤 惟 江汉大学

沈 洁 扬州大学

陈文强 福建工程学院

肖建华 湖南工程学院

邵祖华 浙江科技学院

靳 敏 黑龙江工程学院

序

工程科学技术在推动人类文明的进步中一直起着发动机的作用。随着知识经济时代的到来,科学技术突飞猛进,国际竞争日趋激烈。特别是随着经济全球化发展和我国加入WTO,世界制造业将逐步向我国转移。有人认为,我国将成为世界的“制造中心”。有鉴于此,工程教育的发展也因此面临着新的机遇和挑战。

迄今为止,我国高等工程教育已为经济战线培养了数百万专门人才,为经济的发展作出了巨大的贡献。但据IMD 1998年的调查,我国“人才市场上是否有充足的合格工程师”指标排名世界第36位,与我国科技人员总数排名世界第一形成很大的反差。这说明符合企业需要的工程技术人员特别是工程应用型技术人才市场供给不足。在此形势下,国家教育部近年来批准组建了一批以培养工程应用型本科人才为主的高等院校,并于2001年、2002年两次举办了“应用型本科人才培养模式研讨会”,对工程应用型本科教育的办学思想和发展定位作了初步探讨。本系列教材就是在这种形势下组织编写的,以适应经济、社会发展对工程教育的新要求,满足高素质、强能力的工程应用型本科人才培养的需要。

航天工程的先驱、美国加州理工学院的冯·卡门教授有句名言:“科学家研究已有的世界,工程师创造未有的世界。”科学在于探索客观世界中存在的客观规律,所以科学强调分析,强调结论的惟一性。工程是人们综合应用科学(包括自然科学、技术科学和社会科学)理论和技术手段去改造客观世界的实践活动,所以它强调综合,强调方案优缺点的比较并做出论证和判断。这就是科学与工程的主要不同之处。这也就要求我们对工程应用型人才的培养和对科学研究型人才的培养应实施不同的培养方案,采用不同的培养模式,采用具有不同特点的教材。然而,我国目前的工程教育没有注意到这一点,而是:①过分侧重工程科学(分析)方面,轻视了工程实际训练方面,重理论,轻实践,没有足够的工程实践训练,工程教育的“学术化”倾向形成了“课题训练”的偏软现象,导致学生动手能力差。②人才培养模式、规格比较单一,课程结构不合理,知识面过窄,导致知识结构单一,所学知识中有一些内容已陈旧,交叉学科、信息学科的内容知之甚少,人文社会科学知识薄弱,学生创新能力不强。③教材单一,注重工程的科学分析,轻视工程实践能力的培养;注重理论知识的传授,轻视学生个性特别是创新精神的培养;注重教材的系统性和完整性,造成课程方面的相互重复、脱节等现象;缺乏工程应用背景,存在内容陈旧的现象。④老师缺乏工程实践经验,自身缺乏“工程训练”。⑤工程教育在实践中与经济、产业的联系不密切。要使我国工程教育适应经济、社会的发展,培养更多优秀的工程技术人才,我们必须努力改革。

组织编写本套系列教材,目的在于改革传统的高等工程教育教材,建设一套富有特色、有利于应用型人才培养的本科教材,满足工程应用型人才培养的要求。

本套系列教材的建设原则是:

1. 保证基础,确保后劲

科技的发展,要求工程技术人员必须具备终生学习的能力。为此,从内容安排上,保证学生有较厚实的基础,满足本科教学的基本要求,使学生日后具有较强的发展后劲。

2. 突出特色,强化应用

围绕培养目标,以工程应用为背景,通过理论与工程实际相结合,构建工程应用型本科教育系列教材特色。本套系列教材的内容、结构遵循如下9字方针:知识新、结构新、重应用。教材内容的要求概括为:“精”、“新”、“广”、“用”。“精”指在融会贯通教学内容的基础上,挑选出最基本的内容、方法及典型应用;“新”指将本学科前沿的新进展和有关的技术进步新成果、新应用等纳入教学内容,以适应科学技术发展的需要。妥善处理好传统内容的继承与现代内容的引进。用现代的思想、观点和方法重新认识基础内容和引入现代科技的新内容,并将这些按新的教学系统重新组织;“广”指在保持本学科基本体系下,处理好与相邻以及交叉学科的关系;“用”指注重理论与实际融会贯通,特别是注入工程意识,包括经济、质量、环境等诸多因素对工程的影响。

3. 抓住重点,合理配套

工程应用型本科教育系列教材的重点是专业课(专业基础课、专业课)教材的建设,并做好与理论课教材建设同步的实践教材的建设,力争做好与之配套的电子教材的建设。

4. 精选编者,确保质量

遴选一批既具有丰富的工程实践经验,又具有丰富的教学实践经验的教师担任编写任务,以确保教材质量。

我们相信,本套系列教材的出版,对我国工程应用型人才培养质量的提高,必将产生积极作用,为我国经济建设和社会发展作出一定的贡献。

机械工业出版社颇具魄力和眼光,高瞻远瞩,及时提出并组织编写这套系列教材,他们为编好这套系列教材做了认真细致的工作,并为该套系列教材的出版提供了许多有利的条件,在此深表衷心感谢!

编 委 会 主 任

刘国荣教授

湖南工程学院院长

前 言

数据结构课程在计算机学科学生的培养中地位十分重要。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现实世界中数据的各种逻辑结构、在计算机中的存储结构以及各种操作的算法设计问题。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,从而为进一步学习后续专业课程、为以后进行软件开发和应用打下坚实的基础。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、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、二叉树、图等基本数据结构,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,循序渐进,思路清晰。书中示例选题合适,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 C++ 语言描述数据结构

本书采用 C++ 语言,以面向对象的思想、观点和方法,将主要的抽象数据类型设计成类,包括顺序表类、链表类、顺序串类、栈类、队列类、矩阵类、二叉树类、哈希表类、图类等,并实现了每种抽象数据类型的数据操作算法。

3. 具有丰富翔实的例程

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应用实习题等实践环节。这些为巩固学生的理论知识,提高学生的软件设计能力的培养起到了辅助作用,为以后学生进行系统软件和应用软件的开发研究打下了坚实的基础。

本书所有程序均在 Visual C++ 6.0 中调试通过。

根据作者的教学体会,使用本教材授课约需 64~80 学时。

本书由叶核亚主编,黄纬为副主编,孙水华参编。各章编写分工为:叶核亚编写第 1、3、7、8、9、10 章,黄纬编写第 2、5 章,孙水华编写第 4、6 章。全书由叶核亚修改定稿。本书由朱战立教授主审。朱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。

VIII

由于作者水平有限，时间紧迫，书中难免有疏漏之处，敬请读者批评指正。

E-mail 地址: yeheya@x263.net

作 者

2004 年 6 月

目 录

序 前言

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	4
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	9
1.2.3 算法分析	10
习题1	12
实习1	12
第2章 线性表	14
2.1 线性表的概念	14
2.1.1 线性表的抽象数据类型	14
2.1.2 线性表的存储结构	15
2.2 顺序表类	16
2.2.1 顺序表类声明	16
2.2.2 顺序表类操作	16
2.2.3 顺序表类操作的效率分析	21
2.3 单链表类	23
2.3.1 单链表的概念	23
2.3.2 单链表的节点类	24
2.3.3 单链表类的设计与实现	25
2.3.4 两种存储结构性能的比较	34
2.3.5 单向循环链表类	34
2.4 双向链表类	39
2.4.1 双向链表的概念	39
2.4.2 双向链表的节点类	39
2.4.3 双向链表类的设计与实现	40
2.4.4 双向循环链表的概念	43

习题2	44
实习2	45
第3章 排序	46
3.1 排序的基本概念	46
3.2 插入排序	47
3.2.1 顺序表的直接插入排序	47
3.2.2 单链表的直接插入排序	50
3.2.3 希尔排序	52
3.3 交换排序	55
3.3.1 冒泡排序	56
3.3.2 改进的冒泡排序	57
3.3.3 快速排序	59
3.4 选择排序	62
3.4.1 顺序表的直接选择排序	62
3.4.2 单链表的直接选择排序	64
3.5 归并排序	67
3.5.1 顺序表的归并排序	68
3.5.2 单链表的归并排序	71
习题3	74
实习3	75
第4章 串	77
4.1 串的基本概念	77
4.1.1 串的定义	77
4.1.2 串的抽象数据类型	78
4.1.3 串的存储结构	78
4.2 顺序串类	79
4.2.1 顺序串类声明	79
4.2.2 顺序串类的基本操作	80
4.2.3 顺序串类的运算符重载	83
4.2.4 对子串的操作	85
4.3 串的模式匹配算法	88

4.3.1 Brute-Force 算法	89	第 7 章 树和二叉树	147
4.3.2 替换子串操作	91	7.1 树	147
习题 4	92	7.1.1 树的定义	147
实习 4	92	7.1.2 树的术语	148
第 5 章 栈和队列	93	7.1.3 树的表示方法	149
5.1 栈	93	7.2 二叉树	150
5.1.1 栈的定义	93	7.2.1 二叉树的定义	150
5.1.2 栈的抽象数据类型	94	7.2.2 二叉树的性质	151
5.1.3 顺序栈类	94	7.2.3 二叉树的抽象数据类型	152
5.1.4 链式栈类	98	7.2.4 二叉树的遍历	153
5.1.5 栈的应用	102	7.2.5 二叉树的存储结构	154
5.2 队列	110	7.2.6 树与二叉树的转换	155
5.2.1 队列的定义	110	7.3 二叉树类	156
5.2.2 队列的抽象数据类型	111	7.3.1 二叉树的节点类	156
5.2.3 队列的存储结构	111	7.3.2 二叉树类的设计与实现	158
5.2.4 顺序循环队列类	113	7.3.3 建立二叉树的算法设计	164
5.2.5 链式队列类	116	7.3.4 二叉树遍历的非递归算法	170
5.2.6 队列的应用	119	7.3.5 二叉树的层次遍历	172
5.3 递归	121	7.4 线索二叉树	173
习题 5	126	7.4.1 线索二叉树的定义	174
实习 5	126	7.4.2 线索二叉树的节点类	175
第 6 章 数组和广义表	128	7.4.3 中序线索二叉树类	175
6.1 数组	128	7.5 堆排序	183
6.1.1 一维数组	128	习题 7	187
6.1.2 多维数组	129	实习 7	188
6.2 矩阵类	131	第 8 章 查找	190
6.2.1 矩阵类的声明	131	8.1 查找的基本概念	190
6.2.2 矩阵类的操作	132	8.2 线性表的查找	192
6.3 特殊矩阵的压缩存储	136	8.2.1 顺序查找	192
6.4 稀疏矩阵	136	8.2.2 折半查找	195
6.4.1 稀疏矩阵的三元组线性表	136	8.2.3 分块查找	198
6.4.2 三元组顺序表类	137	8.3 二叉排序树及其查找算法	200
6.4.3 三元组链表	141	8.4 哈希查找	206
6.5 广义表	142	8.4.1 哈希表的基本概念	206
6.5.1 广义表的概念	142	8.4.2 设计哈希函数	208
6.5.2 广义表的存储结构	144	8.4.3 解决冲突的方法	209
习题 6	145	8.4.4 拉链法的哈希表类	210
实习 6	146	习题 8	214

实习 8	215	9.3.2 广度优先遍历	224
第 9 章 图	216	9.4 邻接矩阵图类	224
9.1 图的基本知识	216	9.5 最小生成树	231
9.1.1 图的定义	216	9.5.1 树与图	231
9.1.2 节点的度	218	9.5.2 生成树	232
9.1.3 子图	218	9.5.3 最小生成树及其构造算法	233
9.1.4 路径、回路及连通性	219	9.6 最短路径	235
9.1.5 图的抽象数据类型	220	习题 9	236
9.2 图的存储结构	220	实习 9	237
9.2.1 邻接矩阵	220	第 10 章 综合应用设计	238
9.2.2 邻接表	222	10.1 用“预见算法”解骑士游历问题 ...	238
9.3 图的遍历	223	10.2 综合应用实习	246
9.3.1 深度优先遍历	223	参考文献	250

第1章 绪 论

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。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,如何设计好的算法,就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设计的核心。在计算机领域流传着一句经典名言“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、数据元素、数据类型、抽象数据类型、数据结构、算法等基本概念,以及算法设计与分析的基本方法。

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、数据元素、数据项

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)。数据是计算机程序的处理对象。例如,学生成绩管理程序处理的数据是每个学生的信息,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作一个**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作该数据元素的**数据项**(data item),数据项是数据元素的不可分割的最小单位。

2. 数据类型

类型(type)是一组值的集合。**数据类型**(data type)是指一个类型和定义在这个类型上的操作集合。

数据类型定义了数据元素的性质、取值范围以及对数据元素所能进行的各种操作。例

如, C++语言中整数类型 `int` 的值集是 $\{-2^{32}, \dots, -2, -1, 0, 1, 2, \dots, 2^{32}-1\}$, 除了整数本身的性质外, 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 C++语言中有 `int`、`long`、`float`、`double`、`char` 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数据类型。例如, 学生姓名可以用字符数组表示, 年龄可以用整数类型 `int` 表示, 成绩可以用浮点类型 `float` 表示, 等等。而包括学号、姓名、成绩等的学生信息就是一种复杂数据类型。表 1-1 是一个有 3 个数据元素的学生信息表。其中, 学生的学号、姓名、性别、年龄等数据构成学生信息描述的数据项; 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 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

本书采用 C++语言描述。表 1-1 学生信息表中的数据元素可以声明为如下 `Student` 类的对象。

```
class Student
{
    char number[8];
    char name[20];
    char boy[4];
    int age;
};
```

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这个类型上的操作集合。

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。

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的组织和有效处理。

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

互关联的数据元素之间的关系，关注的是数据元素之间的关系是怎样的。

例如，由一个班级的学生组成的一批数据元素，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；而祖父、父亲、我、儿子、孙子间的辈份关系也是层次关系。这种关系不因数据的改变而改变。

数据结构课程主要讨论线性表、栈、队列、串、数组、树、二叉树、图等典型的常用数据结构，这些典型的常用数据结构就是一个个不同的抽象数据类型。

对于每一种数据结构，都需要从以下3方面来讨论问题：

- 数据的逻辑结构
- 数据的存储结构
- 数据的操作

1.1.2 数据的逻辑结构

数据元素之间的相互联系方式称为**数据的逻辑结构**。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。其中，树结构和图结构是非线性结构。

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 1-1a 所示，其中，数据元素 B 有一个前驱数据元素 A，一个后继数据元素 C。

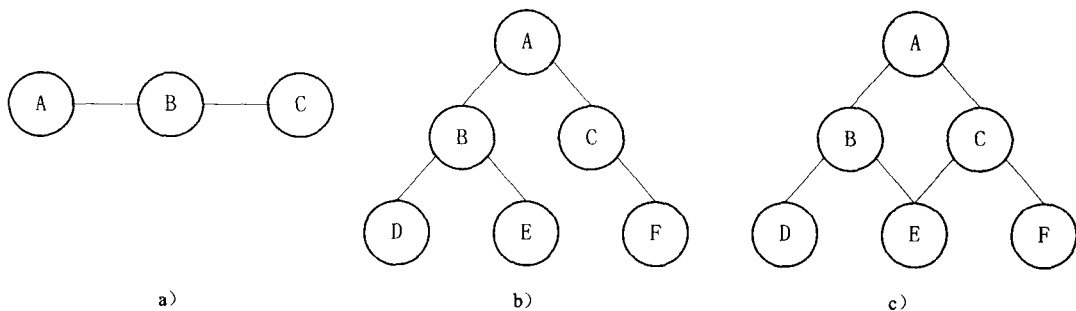


图 1-1 3 种基本的数据结构

a) 线性结构 b) 树结构 c) 图结构

2. 树结构

树结构的一般定义是：除根节点外，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，根节点没有前驱数据元素。

树结构如图 1-1b 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，零个或若干个后继数据元素。

图结构如图 1-1c 所示，其中，数据元素 E 有两个前驱数据元素 B 和 C。

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，连线表示数据元素之间的关系。

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内存中的实现，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两种：顺序存储结构和链式存储结构。

(1) 顺序存储结构 顺序存储结构是把数据元素存储在一块连续地址空间的内存中，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）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次序体现线性结构数据元素之间的逻辑次序。

(2) 链式存储结构 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节点（node）。链式存储结构是使用指针把相互直接关联的节点（即直接前驱节点或直接后继节点）链接起来，其特点是逻辑上相邻的数据元素在物理上不一定相邻，数据间的逻辑关系表现在节点的链接关系上。

对于数据元素是 {A, B, C, D} 的线性表，其顺序和链式存储结构如图 1-2 所示。

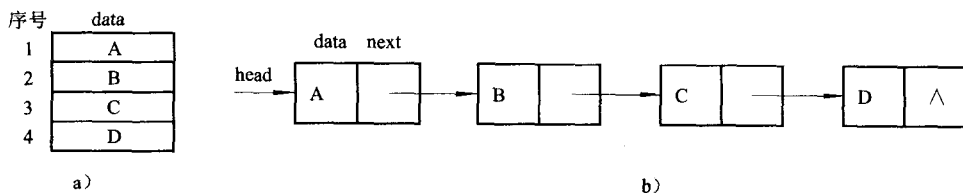


图 1-2 两种存储结构

a) 顺序存储结构 b) 链式存储结构

图 1-2b 以链式存储结构存储的线性表称为链表，就像自行车的链条是由每一节链串起来一样，节点在链表中的位置可根据需要重新组织形成新的链表。

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，每个节点除了要保存数据元素外，还要保存指向后继节点的链信息。因此，每个节点至少由两部分组成：数据域 data —— 保存数据元素，指针域 next —— 指向后继节点。

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。

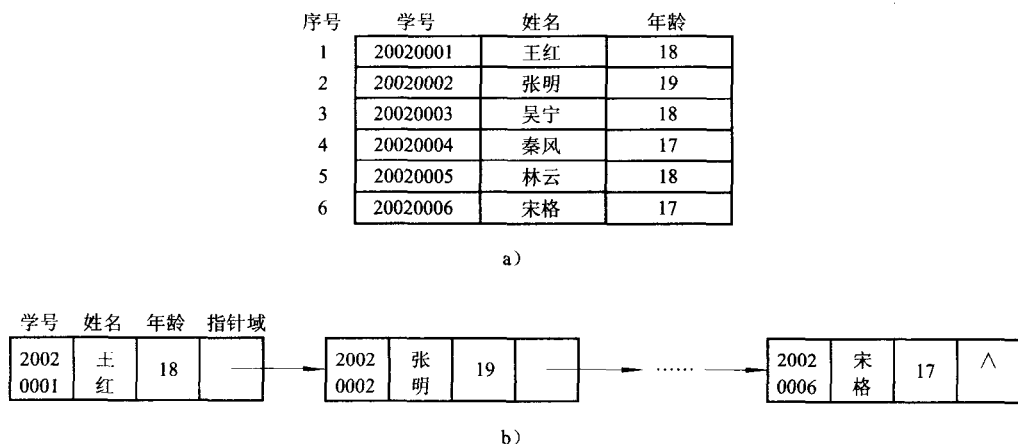


图 1-3 学生信息表的两种存储结构

a) 顺序存储结构 b) 链式存储结构

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。

结构的存储密度定义为数据元素本身所占的存储量和整个结构所占的存储量之比, 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。

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元素进行的某种处理称作**数据的操作**, 对一种数据类型的数据元素进行的所有操作称作**数据的操作集合**。