

● 普通高等教育“十三五”规划教材
(计算机专业群)

数据结构

主 编 韩利凯 朱浩悦
副主编 高寅生 杨彩霖



中国水利水电出版社
www.waterpub.com.cn

普通高等教育“十三五”规划教材（计算机专业群）

数据结构

主 编 韩利凯 朱浩悦

副主编 高寅生 杨彩霖



中国水利水电出版社

www.waterpub.com.cn

· 北京 ·

内 容 提 要

本书系统地介绍了各种常用的数据结构与算法，共分为 10 章。第 1 章为绪论，引入数据结构与算法的一些基本概念；第 2~7 章分别介绍线性表、栈与队列、串、数组与广义表、树与二叉树和图等几种基本的数据结构；第 8~9 章分别介绍各种查找和内、外排序的基本运算，它们都是数据处理中广泛使用的技术；第 10 章列举了数据结构课程设计实例，通过综合训练，培养学生分析问题、解决问题、编程和动手操作等多方面的能力，并运用所学的数据结构知识去解决实际问题。

本书内容丰富，逻辑性强，选材精练，图文并茂，对基本理论的叙述深入浅出、通俗易懂；既注重理论知识，又强调工程实用；精选的应用实例涉及领域相当广泛，给读者提供了思路与方法，有助于提高读者分析问题和解决问题的能力。全书采用类 C 语言作为数据结构和算法的描述语言，每章后进行小结并配有适量习题，便于读者掌握各章的重点和难点并进行必要的训练，既便于教学，又便于自学。

本书可作为计算机类专业或信息类相关专业的本科或专科教材，也可供从事计算机工程与应用工作的科技工作者参考。

图书在版编目（C I P）数据

数据结构 / 韩利凯，朱浩悦主编. -- 北京：中国水利水电出版社，2016.8
普通高等教育“十三五”规划教材. 计算机专业群
ISBN 978-7-5170-4541-0

I. ①数… II. ①韩… ②朱… III. ①数据结构—高等学校—教材 IV. ①TP311.12

中国版本图书馆CIP数据核字(2016)第162427号

策划编辑：石永峰 责任编辑：李 炎 加工编辑：周益丹 封面设计：李 佳

书 名	普通高等教育“十三五”规划教材（计算机专业群） 数据结构 SHUJYU JIEGOU
作 者	主 编 韩利凯 朱浩悦 副主编 高寅生 杨彩霖
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路 1 号 D 座 100038） 网址：www.waterpub.com.cn E-mail: mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）68367658（营销中心）、82562819（万水）
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	三河市鑫金马印装有限公司
规 格	184mm×260mm 16 开本 14.25 印张 349 千字
版 次	2016 年 8 月第 1 版 2016 年 8 月第 1 次印刷
印 数	0001—3000 册
定 价	30.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换
版权所有·侵权必究

前 言

“数据结构”是计算机科学中一门综合性的专业基础课。主要介绍如何合理地组织数据，有效地存储和处理数据，正确地设计算法以及对算法的分析和评价。通过对本课程的学习，使学生深入透彻地理解数据结构的逻辑结构和物理结构的基本概念以及有关算法，培养基本的、良好的程序设计技能，编制高效可靠的程序，为学习操作系统、编译原理和数据库等课程奠定基础。

目前，“数据结构”不仅是计算机专业的一门重要的专业基础课程，而且大多数高等院校的非计算机专业也将“数据结构”作为主干课程。“数据结构”不仅是计算机专业考研的必考科目之一，还是全国计算机等级考试、软件资格（水平）考试的主要考试内容。

本书内容丰富，逻辑性强，选材精练，图文并茂，对基本理论的叙述深入浅出、通俗易懂，既注重理论知识，又强调工程实用，所精选的应用实例涉及领域相当广泛，给读者提供了思路与方法，有助于提高读者分析问题和解决问题的能力。全书采用类 C 语言作为数据结构和算法的描述语言，每章后进行小结并配有适量习题，便于读者掌握各章的重点和难点并进行必要的训练，既便于教学，又便于自学。

本书主要内容如下：

第 1 章：如果读者刚接触数据结构，这一章将告诉您数据结构是什么，以及本书的学习目标、学习方法和学习内容；另外，还介绍了本书对算法的描述方法。

第 2 章：主要介绍了线性表。首先讲解线性表的逻辑结构，然后介绍线性表的各种常用存储结构，在每一节均给出了算法的具体应用。通过这一章的学习，读者可以掌握顺序表、动态链表和静态链表的操作。

第 3 章：主要介绍操作受限的线性表——栈和队列，内容包括栈的定义，栈的基本操作及栈与递归的转化，队列的概念，顺序队列和链式队列的运算。

第 4 章：主要介绍一种特殊的线性表——串，内容包括串的概念，串的各种存储表示，以及串的模式匹配算法。

第 5 章：主要介绍数组与广义表，内容包括数组的概念，数组（矩阵）的存储结构及运算，几种特殊矩阵；广义表的概念，广义表的两种存储方式，广义表的操作实现。

第 6 章：主要介绍非线性数据结构——树和二叉树，内容包括树和二叉树的概念，树和二叉树的存储表示，二叉树的性质，二叉树的遍历和线索化，树、森林与二叉树的转换及哈夫曼树。

第 7 章：主要介绍非线性数据结构——图，内容包括图的概念和存储结构、图的遍历、最小生成树、拓扑排序、关键路径及最短路径。

第 8 章：主要介绍数据结构的常用技术——查找。首先介绍查找的概念，然后结合具体实例介绍各种查找算法，并给出了完整程序。

第 9 章：主要介绍数据结构的常用技术——排序。首先介绍排序的相关概念，然后介绍各种内外排序技术，并给出了具体实现算法。

第 10 章：通过对本章的学习，可以复习数据结构的基本算法，学习数据结构实验系统开发的全过程，了解系统设计的一般要求；掌握文件包含处理的基本方法；提高程序编写和程序调试的能力。

本书由西安文理学院韩利凯、朱浩悦担任主编，高寅生、杨彩霖担任副主编。其中第 1 章由李向军编写；第 2 章、第 3 章、第 7 章由韩旭编写；第 4 章、第 8 章由朱浩悦编写；第 5 章、第 6 章由王昭阳编写；第 9 章由刘光军、韩利凯编写；第 10 章由韩利凯编写。

在教材编写过程中，编者参考了国内外众多数据结构与算法方面的优秀教材，其中大多列举在书后的参考文献中。在此，我们对这些教材的编著者表示衷心感谢。

中国水利水电出版社的领导和编审老师给予本书编写以很大的支持与细心指导，在此也表示衷心的感谢！

由于编者水平有限，时间仓促，书中难免存在一些不足之处，殷切希望广大读者批评指正。

编 者

2016 年 6 月

目 录

前言

第1章 绪论	1
1.1 数据结构的研究内容	1
1.2 基本概念和术语	4
1.2.1 数据、数据元素、数据项和数据对象	4
1.2.2 数据结构	4
1.2.3 数据类型和抽象数据类型	8
1.3 算法与数据结构	12
1.3.1 数据结构上的基本运算	13
1.3.2 算法的定义及特性	14
1.3.3 算法分析	14
1.4 本章小结	25
习题1	26
第2章 线性表	29
2.1 线性表的概念及其抽象数据类型定义	29
2.1.1 线性表的逻辑结构	29
2.1.2 线性表的抽象数据类型定义	30
2.2 线性表的顺序存储	31
2.2.1 线性表的顺序存储结构	31
2.2.2 线性表顺序存储结构上的基本运算	33
2.3 线性表的链式存储	36
2.3.1 单链表	36
2.3.2 单链表上的基本运算	37
2.3.3 循环链表	43
2.3.4 双向链表	44
2.3.5 静态链表	46
2.4 线性表应用——一元多项式的表示及相加	47
2.5 顺序表与链表的综合比较	50
2.5.1 顺序表和链表的比较	50
2.5.2 线性表链式存储方式的比较	51
2.6 本章小结	51
习题2	52

第3章 限定性线性表——栈与队列	55
3.1 栈	55
3.1.1 栈的定义	55
3.1.2 栈的表示和实现	57
3.1.3 栈的应用举例	61
3.2 队列	63
3.2.1 队列的定义	63
3.2.2 队列的表示和实现	65
3.2.3 队列的应用举例	70
3.3 本章小结	73
习题3	74
第4章 串	76
4.1 串的基本概念	76
4.1.1 串的定义	76
4.1.2 串的基本操作	77
4.2 串的存储实现	79
4.2.1 定长顺序串表示	79
4.2.2 堆分配存储表示	82
4.2.3 串的链接存储表示	84
4.2.4 子串定位运算的实现	86
4.3 串的应用举例	88
4.4 本章小结	89
习题4	90
第5章 数组和广义表	92
5.1 数组的定义与基本操作	92
5.1.1 数组的定义	92
5.1.2 数组的基本操作	93
5.2 数组的顺序存储和实现	93
5.2.1 数组的顺序存储结构	93
5.2.2 数组的顺序存储的实现	95
5.3 特殊矩阵的压缩存储	96
5.3.1 规律分布的特殊矩阵	96
5.3.2 稀疏矩阵	98

5.4 广义表·····	100	7.4.1 图的连通性问题·····	152
5.4.1 广义表的概念·····	101	7.4.2 有向无环图的应用·····	156
5.4.2 广义表的存储结构·····	102	7.4.3 最短路径问题·····	160
5.4.3 广义表的操作实现·····	104	7.5 本章小结·····	162
5.5 本章小结·····	106	习题 7·····	162
习题 5·····	106	第 8 章 查找 ·····	164
第 6 章 树与二叉树 ·····	109	8.1 查找的基本概念·····	164
6.1 树的定义与基本术语·····	109	8.2 顺序查找法·····	165
6.2 二叉树·····	111	8.3 折半查找法·····	166
6.2.1 二叉树的定义与基本操作·····	112	8.4 散列表及其查找·····	169
6.2.2 二叉树的性质·····	112	8.4.1 常用的散列函数·····	170
6.2.3 二叉树的存储结构·····	114	8.4.2 处理冲突的方法·····	171
6.3 二叉树的遍历与线索化·····	116	8.4.3 散列表的查找分析·····	175
6.3.1 二叉树的遍历·····	116	8.5 本章小结·····	177
6.3.2 遍历算法应用·····	119	习题 8·····	177
6.3.3 线索二叉树·····	122	第 9 章 内部排序 ·····	179
6.3.4 由遍历序列确定二叉树·····	126	9.1 排序的概念·····	179
6.4 树、森林和二叉树的关系·····	127	9.1.1 排序的稳定性·····	179
6.4.1 树的存储结构·····	127	9.1.2 排序用到的结构与操作·····	179
6.4.2 树、森林与二叉树的相互转换·····	129	9.2 插入排序·····	180
6.4.3 树与森林的遍历·····	132	9.2.1 直接插入排序·····	180
6.5 哈夫曼树及其应用·····	134	9.2.2 直接插入排序复杂度分析·····	181
6.5.1 哈夫曼树·····	134	9.2.3 其他插入排序·····	181
6.5.2 哈夫曼编码·····	136	9.3 交换排序·····	184
6.6 本章小结·····	138	9.3.1 最简单的交换排序·····	184
习题 6·····	138	9.3.2 冒泡排序·····	184
第 7 章 图 ·····	141	9.3.3 冒泡排序的优化·····	185
7.1 图的定义与基本术语·····	141	9.3.4 快速排序·····	186
7.1.1 图的定义·····	141	9.4 选择排序·····	189
7.1.2 图的基本术语·····	141	9.4.1 简单选择排序·····	189
7.2 图的存储结构·····	143	9.4.2 树形选择排序·····	190
7.2.1 邻接矩阵表示法·····	143	9.4.3 堆排序·····	191
7.2.2 邻接表表示法·····	145	9.5 归并排序·····	196
7.2.3 十字链表·····	146	9.6 基数排序·····	196
7.2.4 邻接多重表·····	148	9.6.1 多关键字排序·····	196
7.3 图的遍历·····	149	9.6.2 链式基数排序·····	197
7.3.1 深度优先搜索·····	149	9.7 内部排序算法的分析和比较·····	198
7.3.2 广度优先搜索·····	150	9.8 本章小结·····	199
7.4 图的应用·····	152	习题 9·····	200

第 10 章 数据结构课程实训·····	202	10.2.3 “数据结构实验系统”主控模块···	209
10.1 系统设计的要求·····	202	10.3 数据结构课程设计·····	210
10.1.1 系统总体结构设计·····	202	10.3.1 课程设计的目的·····	210
10.1.2 系统详细设计·····	204	10.3.2 课程设计的内容·····	210
10.1.3 系统调试和维护·····	205	10.3.3 课程设计报告·····	211
10.2 文件的包含处理·····	206	10.3.4 课程设计的考核·····	212
10.2.1 什么是文件包含·····	206	10.4 课程设计的要求·····	212
10.2.2 如何对“数据结构实验系统” 进行文件包含·····	208	10.5 课程设计题目·····	213
		参考文献·····	220

第 1 章 绪论

早期的计算机主要用于数值计算，现在，计算机主要用于非数值计算，包括处理字符、表格和图像等具有一定结构的数据。这些数据内容存在着某种关系，只有明确数据的内在关系，合理地组织数据，才能对数据进行有效的处理，设计出高效的算法。如何合理地组织数据、高效地处理数据，这就是“数据结构”主要研究的问题。本章简要介绍有关数据结构的基本概念及基本的算法分析与设计方法。

1.1 数据结构的研究内容

计算机主要用于数值计算时，一般要经过如下几个步骤：首先从具体问题抽象出数学模型，然后设计一个求解此数学模型的算法，最后编写程序，进行测试、调试，直到解决问题。在此过程中寻求数学模型的实质是分析问题，从中提取操作的对象，并找出这些操作对象之间的关系，然后以数学语言加以描述，即建立相应的数学模型。例如，用计算机进行全球天气预报时，就要求解一组球面坐标系下的二阶椭圆偏微分方程；预测人口增长情况的数学模型为常微分方程。求解这些数学方程的算法属于计算数学研究的范畴，如高斯消元法、差分法、有限元法等算法。数据结构主要研究非数值计算问题。通常情况下，非数值计算问题无法用数学方程建立数学模型，下面通过三个实例加以说明。

例 1.1 学生学籍管理系统。

目前各高校教务处使用计算机对全校的学生情况作统一管理。要了解学生的基本信息，包括学生的学号、姓名、性别、生源地、专业等，如表 1-1 所示。每个学生的基本情况按照不同的序号依次放在“学生基本信息表”中，根据需要可对这张表进行查找、插入及删除等相关操作。表中每个学生的基本信息记录按序号排列，形成了学生基本信息记录的线性序列，呈现出一种线性关系。

表 1-1 学生基本信息表

学号	姓名	性别	生源地	专业
1401130118	马尧	男	陕西西安	计算机科学与技术
1401130119	欧阳雪涛	男	河南南阳	计算机科学与技术
1401130120	史宇飞	男	河北石家庄	计算机科学与技术
1401130121	孙甜珊	女	宁夏石嘴山	计算机科学与技术
1401130122	汪一景	女	陕西宝鸡	计算机科学与技术
1401130123	王浩然	男	宁夏石嘴山	计算机科学与技术
0204140316	刘笑	女	甘肃天水	物联网工程
0204140321	屈航	男	江苏镇江	物联网工程

续表

学号	姓名	性别	生源地	专业
0204140324	史浩	男	陕西铜川	物联网工程
0204140328	王芮	女	河南开封	物联网工程
0204140331	吴奇奇	男	宁夏固原	物联网工程
0204140333	熊轩晖	男	青海西宁	物联网工程

类似的线性表结构还有图书馆的书目管理系统、库存管理系统等。在这类问题中，计算机处理的对象是各种表，元素之间存在着简单的一对一的线性关系，因此这类问题的数学模型就是各种表，可施加于对象上的操作有查找、插入和删除等。这类数学模型也称为“线性”的数据结构。

例 1.2 人机对弈问题。

英国公司 DeepMind 研发的 AlphaGo 与韩国棋手李世石的人机大战成为近期的热门话题，也引发了更多的人关注人工智能领域的最新研究进展。人机对弈是人工智能领域的一个重要研究方向，计算机之所以能和对弈是因为实现已经将对弈的策略存储在了计算机中。由于对弈的过程是在一定规则下随机进行的，所以，为使计算机能够灵活对弈，就必须把对弈过程中所有可能发生的情况及相应的对策都加以考虑。这里以最简单的井字棋为例，其初始状态是一个空的棋盘格局。对弈开始后，每下一步棋，则构成一个新的棋盘格局，且相对于上一个棋盘格局的可能选择可以有多种形式，因而整个对弈过程就像图 1-1 所示的“一棵倒长的树”。在这棵“树”中，从初始状态（树根）到某一最终格局（叶子）的一条路径，就是一次具体的对弈过程。

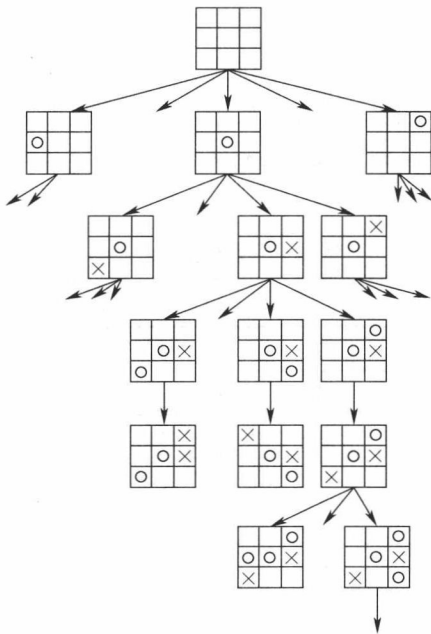


图 1-1 井字棋的对弈树

类似的树型结构还有计算机的文件系统、一个单位的组织机构等。在这类问题中，计算机处理的对象就是树结构，元素之间是一种一对多的层次关系，可施加于对象上的操作有查找、插入和删除等。人机对弈问题的数学模型就是如何用树型结构表示棋盘和棋子等，算法是博弈的规则和策略。这类数学模型称为“树”的数据结构。

例 1.3 最短路径问题。

假设从城市 A 到城市 B 有多条线路，但每条线路的交通费不同，那么，如何选择一条线路，使从城市 A 到城市 B 的交通费用最少。解决的方法是，可以将此类问题抽象为图的最短路径问题。如图 1-2 所示，图中的顶点代表城市，有向边代表两个城市之间的通路，边上的权值代表两个城市之间的交通费。求解 A 到 B 的最少交通费用，就是要在有向图中 A 点（起点）到达 B 点（终点）的多条路径中，寻找一条各边权值之和最小的路径，即最短路径。

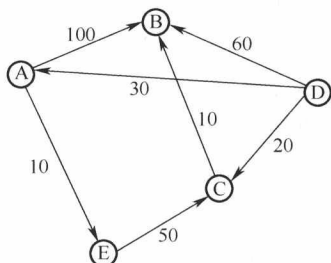


图 1-2 最短路径问题

类似的图型结构还有网络工程图和网络通信图等，这类数学模型称为“图”的数据结构。在这类问题中，元素之间是多对多的网状关系，可施加于对象上的操作依然有查找、插入和删除等。最短路径问题的数学模型就是图结构，算法是求解两点之间的最短路径。

从上面三个实例可以看出，非数值计算问题的数学模型不再是数学方程，而是诸如线性表、树和图的数据结构。因此，可以说，数据结构是一门研究非数值计算程序设计中的操作对象，以及这些对象之间的关系和操作的课程。

“数据结构”作为一门独立的课程在国外是从 1968 年才开始设立的，同年，著名计算机科学家 D.E.Knuth 教授发表了《计算机程序设计艺术》第一卷《基本算法》，这是第一本较系统地阐述“数据结构”基本内容的著作。之后，随着大型程序和大规模文件系统的出现，结构化程序设计成为程序设计方法学的主要研究方向，人们普遍认为程序设计的实质就是对所处理的问题选择一种好的数据结构，并在此结构基础上施加一种好的算法，著名科学家 N.Wirth 教授的“算法+数据结构=程序”正是这种观点的直接体现。

目前，数据结构在计算机科学中是一门综合性的专业基础课。数据结构的研究不仅涉及计算机硬件（特别是编码理论、存储装置和存取方法）的研究范围，而且和计算机软件的研究有着密切的关系，无论是编译程序还是操作系统都涉及数据元素在存储器中的分配问题。在研究信息检索时也必须考虑如何组织数据，以使存取和查找数据元素更为方便。因此，可以说数据结构是介于数学、计算机硬件和软件三者之间的一门核心课程。

有关“数据结构”的研究仍在不断发展。一方面，面向各专门领域中特殊问题的数据结构研究正在不断向前推进；另一方面，从抽象数据类型的观点来讨论数据结构，已成为一种新的趋势，越来越被人们所重视。

1.2 基本概念和术语

1.2.1 数据、数据元素、数据项和数据对象

数据(Data)是客观事物的符号表示,是所有能输入到计算机中并被计算机程序处理的符号的总称。如数学计算中用到的整数和实数,文本编辑中用到的字符串,多媒体程序处理的图形、图像、声音及动画等通过特殊编码以后的数据。

数据元素(Data Element)是数据的基本单位,在计算机中通常作为一个整体进行考虑和处理,有时候也将数据元素称为元素、结点、记录等。数据元素可以完整地描述一个对象,如一名学生记录、对弈树中棋盘的一个格局(状态),以及图中的一个顶点等。

数据项(Data Item)是组成数据元素的、有独立含义的、不可分割的最小单位。例如,学生基本信息表中的学号、姓名、性别、生源地、专业等都是数据项。

数据对象(Data Object)是性质相同的数据元素的集合,是数据的一个子集。例如,整数数据对象是集合 $N=\{0, \pm 1, \pm 2, \dots\}$, 字母字符数据对象是集合 $C=\{'A', 'B', \dots, 'Z', 'a', 'b', \dots, 'z'\}$, 学生基本信息表也可以是一个数据对象。由此可以看出,不论数据元素集合是无限集(如整数集),或是有限集(如字母字符集),还是由多个数据项组成的复合数据元素(如学生表),只要性质相同,都是同一个数据对象。

1.2.2 数据结构

数据结构(Data Structure)是相互之间存在一种或多种特定关系的数据元素的集合,也就是说,数据结构是带“结构”的数据元素的集合,“结构”就是指数据元素之间存在的关系。

数据结构通常包含数据的逻辑结构及数据的物理结构两个层次。

1. 逻辑结构及其描述

数据的逻辑结构是从逻辑关系上描述数据,它与数据的存储无关,是独立于计算机的。因此,数据的逻辑结构可以看作是从具体问题抽象出来的数学模型。

数据的逻辑结构有两个要素:数据元素和关系。数据元素的含义如前所述,关系是指数据元素之间的逻辑关系。根据数据元素之间关系的不同特性,通常有4种基本结构,如图1-3所示。

(1) 集合结构

集合中的数据元素之间的次序是任意的。元素之间除了“属于同一集合”的关系外,没有其他的关系。例如,班级里的每一名学生,某次聚会中的每一个参加者,公共汽车上的所有乘客,存放在仓库中的每件商品。由于集合结构的元素间没有固有关系,因此往往需要借助于其他结构才能在计算机中表示。

(2) 线性结构

数据结构元素之间构成一个有序序列,存在着一对一的关系。其中,第一个元素只有后继没有前驱,最后一个元素只有前驱没有后继。除了第一个和最后一个元素外,其余元素都有一个前驱和一个后继。例如,将学生信息数据按照其学号进行排列,将组成一个线性结构。再如,水泊梁山上的108条好汉,他们形成了一个集合,但他们之间有一个次序,宋江排第一,

卢俊义排第二……，如此下去。

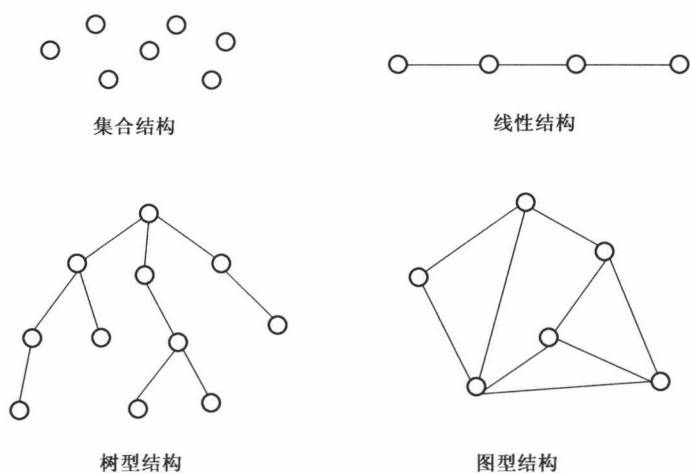


图 1-3 四类基本逻辑结构关系图

(3) 树型结构

树型结构表示的是一种层次关系，树中元素之间存在着一对多的关系。在树型结构中，除了一个特殊的元素（称为根结点或树根、根）外，树中每个元素有且仅有一个前驱，却可以有多个后继，注意根元素没有前驱。树型结构的一个典型的实例就是家族的族谱，一个家族可以以树型结构来表示，即族谱可以看作是一棵树。这家的老祖宗就是族谱树的树根，老祖宗的儿子就是老祖宗的后继。在族谱树中，每个人可以有多个儿子，因此后继数目不限。但每个人只能有一个父亲，因此只有一个前驱。老祖宗的儿子们又可以有他们的儿子，这样就形成了一棵树。老祖宗的父亲无历史记载，因此老祖宗就没有前驱。

(4) 图型结构/网状结构

图是最一般的逻辑结构，图中数据元素之间存在着多对多的关系，每个元素的前驱和后继数目都不限。例如，多位同学之间的朋友关系，任何两位同学都可以是朋友，从而可以构成图型结构或者网状结构。再如，在一个计算机网络中，各个网络设备之间是由线路连接起来的。如果把连接看成是网络设备之间的关系，那么一个计算机网络的拓扑结构就形成了一个图型结构。因为每台网络设备都可以和多台其他设备相连接，向多台设备发送信息，也可以接收多台设备发来的信息。

以上 4 种结构中的集合结构、树型结构及图型结构都属于非线性结构。

线性结构包括：线性表（如学生基本信息表）、栈和队列（具有特殊限定的线性表，数据操作只能在表的一端或两端进行）、字符串（其数据元素仅由单个字符组成）、数组（线性表的推广，其数据元素是一个线性表）、广义表（线性表的推广，其数据元素既可以是单个元素，也可以是一个线性表）。

非线性结构包括：树型结构（具有多个分支的层次结构）、二叉树（具有两个分支的分层结构）、有向图（图型结构，其边是顶点的有序对）、无向图（图型结构，其边是顶点的无序对）。

线性结构和非线性结构构成了数据的逻辑结构，如图 1-4 所示。

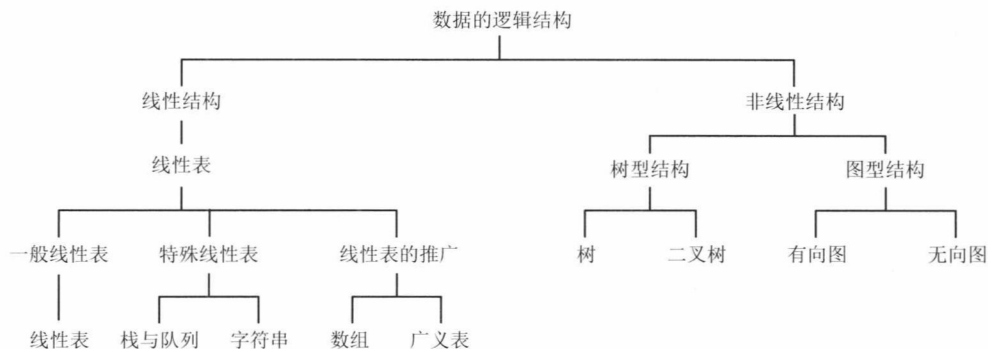


图 1-4 数据逻辑结构层次图

需要注意的是：一些表面上不相同的数据也可以有相同的逻辑结构。例如，一个计算机网络可以用一个图型结构来表示，每个数据元素是一个网络设备，数据元素之间的关系是设备间的物理介质。如果两台设备间由一条物理媒介连接起来，则两台设备间有关系。如果用先导课程和后继课程作为课程间的关系，则一个课程系统中的课程也是一个图型结构。一个课程可以有若干门后继课程。因此，逻辑结构本质上其实就是数据的组织方式。只要解决了这个本质问题，就可以把解决方案用到各种实际应用中，而不必分别对每个问题进行研究。

关于逻辑结构，有以下几点需要特别说明：

1) 逻辑结构与数据元素本身的内容无关。例如，在家谱管理系统中，不管如何描述一个人，人和人之间的关系总是树型关系。当用姓名、性别、出生年月描述一个人时，是树型关系，如果每个人再加上一张照片后，还是树型关系。

2) 逻辑结构与数据元素的个数无关。例如，在家谱中增加或删除某些人，整个家谱还是一个树型结构。

2. 存储结构及其实现

数据存储实现的基本目标是建立数据的机内存储表示方式，它包括两个部分：数据元素的存储和数据元素之间的关系的存储。有时为了方便运算的实现，还可能会增加一些辅助信息的存储。数据对象在计算机中的存储表示称为数据的**存储结构**，也称为**物理结构**。一个物理结构通常由以下三个部分组成：

- 1) 存储结点，每个存储结点存放一个数据元素。
- 2) 数据元素之间的关系的存储，也就是逻辑结构的机内表示。
- 3) 附加信息，为方便运算的实现而设置的一些“哑结点”，如链表中的头结点。

其中，前两个部分是每个存储实现都必须具备的，第三部分则可根据运算实现的需要来决定是否设置。由于数据元素本身比较简单，因此存储结点的组织也比较简单，可能是一个基本的数据类型，也可能是一个记录类型或者用户定义的类型。因此，物理结构主要讨论的是数据元素之间的关系的表示。由于每个数据元素被表示为一个存储结点，所以逻辑结构就由存储结点之间的关联方式间接地表示。

通常数据元素在计算机中有两种基本的存储结构：顺序存储结构和链式存储结构（非顺序存储结构的一种形式）。存储结点之间的关联可以由以下 4 种方式实现。

（1）数据的顺序存储实现（顺序存储结构）

在数据的顺序存储方式中，所有的存储结点存放在一块连续的存储区域中，结点之间的

逻辑关系可以通过结点的存储位置来体现。在高级语言中，一块连续的存储空间通常可以用一个数组来表示。因此，数据的顺序存储实现通常是用一个数据元素类型的数组来存储数据。

顺序存储结构是借助元素在存储器中的相对位置来表示数据元素之间的逻辑关系，通常借助程序设计语言的数组类型来描述。

在前面的“学生基本信息表”中，假定每个结点（学生记录）占用 50 个存储单元，数据从 0 号单元开始由低地址端向高地址端存储，对应的顺序存储结构如表 1-2 所示。

表 1-2 “学生基本信息表”对应的顺序存储结构

地址	学号	姓名	性别	生源地	专业
0	1401130118	马尧	男	陕西西安	计算机科学与技术
50	1401130119	欧阳雪涛	男	河南南阳	计算机科学与技术
100	1401130120	史宇飞	男	河北石家庄	计算机科学与技术
150	1401130121	孙甜珊	女	宁夏石嘴山	计算机科学与技术
200	1401130122	汪一景	女	陕西宝鸡	计算机科学与技术
250	1401130123	王浩然	男	宁夏石嘴山	计算机科学与技术
300	0204140316	刘笑	女	甘肃天水	物联网工程
350	0204140321	屈航	男	江苏镇江	物联网工程
400	0204140324	史浩	男	陕西铜川	物联网工程
450	0204140328	王芮	女	河南开封	物联网工程
500	0204140331	吴奇奇	男	宁夏固原	物联网工程
550	0204140333	熊轩晖	男	青海西宁	物联网工程

(2) 数据的链式存储实现（链式存储结构）

顺序存储结构要求所有的元素依次存放在一片连续的存储空间中，而链式存储结构无需占用一整块存储空间。存储结点可以分散地存放在存储器的不同位置，结点之间的关系通过一个指针显式地指出。因此，在链接存储中，每个存储结点包含两个部分：数据元素部分和指针部分。数据元素部分保存数据元素的值，指针部分保存一组指针，每个指针指向一个与本结点有逻辑关系的结点，即让该指针指向当前元素的后继元素。链式存储结构通常借助于程序语言的指针类型来描述。

假定给前面的“学生基本信息表”中的每个结点附加一个“下一个结点地址”，即后继指针字段用于存放后继结点的首地址，则可以得到如表 1-3 所示的链式存储结构。从表中可以看出，每个结点占用两个连续的存储单元，一个存放结点的信息（学生记录），另一个结点存放后继结点的首地址。

表 1-3 “学生基本信息表”对应的链式存储结构

地址	学号	姓名	性别	生源地	专业	后继结点首地址
0	1401130118	马尧	男	陕西西安	计算机科学与技术	50
50	1401130119	欧阳雪涛	男	河南南阳	计算机科学与技术	100
100	1401130120	史宇飞	男	河北石家庄	计算机科学与技术	150

续表

地址	学号	姓名	性别	生源地	专业	后继结点首地址
150	1401130121	孙甜珊	女	宁夏石嘴山	计算机科学与技术	200
200	1401130122	汪一景	女	陕西宝鸡	计算机科学与技术	250
250	1401130123	王浩然	男	宁夏石嘴山	计算机科学与技术	Null (无后继)
1000	0204140316	刘笑	女	甘肃天水	物联网工程	1050
1050	0204140321	屈航	男	江苏镇江	物联网工程	1100
1100	0204140324	史浩	男	陕西铜川	物联网工程	1150
1150	0204140328	王芮	女	河南开封	物联网工程	1200
1200	0204140331	吴奇奇	男	宁夏固原	物联网工程	1250
1250	0204140333	熊轩晖	男	青海西宁	物联网工程	Null (无后继)

(3) 散列存储方式

数据的散列存储是专用于集合结构的数据存储方式。在散列存储中，各个结点均匀地分布在一块连续的存储区域中，用一个散列函数将数据元素和存储位置关联起来。

(4) 索引存储方式

数据的索引存储是指所有的存储结点按照生成的次序连续存放，另外设置一个索引区域来表示结点之间的关系。

这些基本的存储方式以及它们的组合可以实现数据的灵活存储。

为了更清楚地描述链式存储结构，可以采用更直观的图来表示，如“学生基本信息表”的链式存储结构可用图 1-5 所示的方法表示。

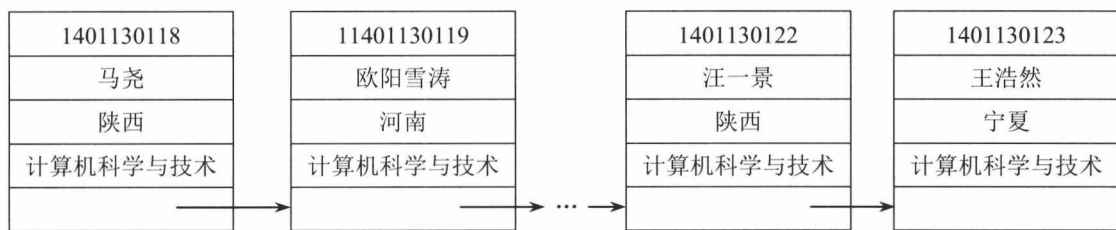


图 1-5 链式存储结构示意图

1.2.3 数据类型和抽象数据类型

1. 数据类型

数据类型 (Data Type) 是高级程序设计语言中的一个基本概念，前面提到过顺序存储结构可以借助程序设计语言的数组类型描述，链式存储结构可以借助指针类型描述，所以数据类型和数据结构的概念密切相关。

一方面，在程序设计语言中，每一个数据都属于某种数据类型。数据类型明显或隐含地规定了数据的取值范围、存储方式及允许进行的运算。数据类型是一个值的集合定义在这个值集上的一组操作的总称。例如，C 语言中的整型变量，其值集为某个区间上的整数（区间大小

依赖于不同的机器), 定义在其上的操作有加、减、乘、除和取模等算术运算; 而实型变量也有其取值范围和相应的运算, 比如整数集上的取模运算是不能用于实数集的。程序设计语言允许用户直接使用的数据类型由具体语言决定, 数据类型反映了程序设计语言的数据描述和处理能力。C 语言除了提供整型、实型及字符型等基本数据类型之外, 还允许用户自定义各种数据类型, 例如数组、结构体和指针等。

2. 抽象数据类型

抽象就是抽取出实际问题的本质。在计算机中使用二进制数来表示数据, 在汇编语言中则给出各种数据的十进制表示, 它们是二进制数据的抽象, 使用者在编程时可以直接使用, 不必考虑实现细节。在高级语言中, 则给出更高一级的数据抽象, 出现了数据类型, 如整型、实型、字符型等, 可以进一步利用这些类型构造出线性表、栈、队列、树、图等复杂的抽象数据类型。

抽象数据类型 (Abstract Data Type, ADT) 一般指由用户定义的, 表示应用问题的数学模型, 以及定义在这个模型上的一组操作的总称, 具体包括三部分: 数据对象, 数据对象上关系的集合, 以及对数据对象的基本操作的集合。

抽象数据类型的定义格如下:

ADT 抽象数据类型名

```
{
    数据对象: <数据对象的定义>
    数据关系: <数据关系的定义>
    基本操作: <基本操作的定义>
}
```

其中, 数据对象和数据关系的定义采用数学符号和自然语言描述, 基本操作的定义格式为:

基本操作名 <参数表>

初始条件: <初始条件描述>

操作结果: <操作结果描述>

基本操作有两种参数: 赋值参数只为操作提供输入值; 引用参数以“&”打头, 除可以提供输入值外, 还将返回操作结果。“初始条件”描述了操作执行之前数据结构和参数应满足的条件, 若初始条件为空, 则省略。“操作结果”说明了操作正常完成之后, 数据结构的变化状况和应返回的结果。

3. 抽象数据类型的表示与实现

抽象数据类型的概念与面向对象方法的思想是一致的。抽象数据类型独立于具体实现, 将数据和操作封装在一起, 使得用户程序只能通过抽象数据类型定义的某些操作来访问其中的数据, 从而实现了信息隐藏。在 C++ 中, 可以用类的声明表示抽象数据类型, 用类的实现来实现抽象数据类型。因此, C++ 中实现的类相当于数据的存储结构及其在存储结构上实现的对数据的操作。

抽象数据类型和类的概念实际上反映了程序设计或软件设计的两层抽象: 抽象数据类型相当于在概念层 (或称为抽象层) 上描述问题, 而类相当于在实现层上描述问题。此外, C++ 中的类只是一个由用户定义的普通类型, 可用它来定义变量 (称为对象或类的实例)。因此, 在 C++ 中, 最终是通过操作对象来解决实际问题的, 所以我们可以将该层次看作是应用层。

由此可以看出, 最终描述和实现抽象数据类型, 最好采用面向对象的方法, 本书基于