

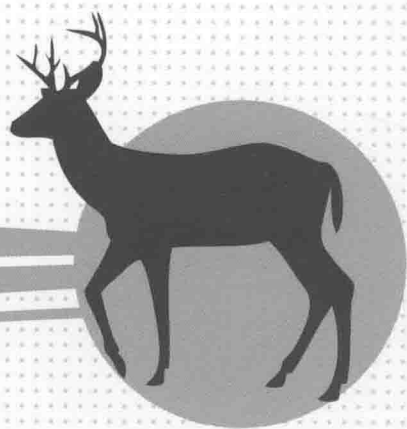


工业和信息化部职业教育指导委员会“十二五”规划教材
全国高等职业教育计算机类规划教材·实例与实训教程系列
国家高职示范院校核心课程教材

实用数据结构

SHIYONG SHUJU JIEGOU

○ 刘 畅 主 编
○ 汤晓伟 王 茹 王象刚 副主编
○ 董凤服 主 审



- ◆ 基础理论知识由浅入深、够用为度, 让学生易学易懂
- ◆ 从实用出发, 通过“数据结构实用系统”贯穿, 培养应用能力
- ◆ 上机实验部分由每章的主要算法构成一个相对独立的子系统
- ◆ 立体配套资源: 电子课件、源代码、教学计划、模拟套题、参考答案



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

全国高等职业教育计算机类规划教材·实例与实训教程系列
国家高职示范院校核心课程教材

实用数据结构

刘 畅 主 编

汤晓伟 王 茹 王象刚 副主编

董凤服 主 审

電 子 工 業 出 版 社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要讨论了线性表、栈、队列、串和广义表、树、图、查找、排序等各种数据结构的典型内容。全书共 10 章, 每章提供了一个体现该章基本内容的实验, 用以验证各章教学的重点算法, 每个实验都有明确的实验目的和实验要求, 用以提高学生自主设计程序的能力。最后一章数据结构实验系统开发, 提出了系统设计的要求、文件的包含处理以及主控模块的设计。每章配有类型丰富的练习题, 并在附录中提供习题参考答案。

本书对数据结构的概念和原理的阐述通俗易懂, 习题的选择难易适当, 题型丰富; 全部算法用 C 语言书写, 对各章子系统的实验和一些重要算法均给出了完整的 C 语言源程序, 并在 VC++ 环境中运行通过。

本书既可作为高职高专院校计算机专业数据结构学科的教科书, 也可以作为成人教育、自学考试和从事计算机应用的工程技术人员的参考书。本教材配有免费的电子课件、教学计划、习题参考答案等配套资源, 可从电子工业出版社的华信教育资源网 (www.hxedu.com.cn) 下载。

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

实用数据结构 / 刘畅主编. —北京: 电子工业出版社, 2011.8

全国高等职业教育计算机类规划教材. 实例与实训教程系列

ISBN 978-7-121-14054-9

I. ①实… II. ①刘… III. ①数据结构—高等职业教育—教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字 (2011) 第 135492 号

策划编辑: 左 雅

责任编辑: 左 雅 特约编辑: 朱英兰

印 刷: 北京季蜂印刷有限公司

装 订: 三河市皇庄路通装订厂

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

开 本: 787×1 092 1/16 印张: 17.75 字数: 454.4 千字

印 次: 2011 年 8 月第 1 次印刷

印 数: 4 000 册 定价: 32.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

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。

服务热线: (010) 88258888。

前 言

数据结构是一门研究各种数据的组织方法、存储方法及在各种数据结构上执行操作的算法的学科。它是计算机专业及其相关专业的一门专业核心基础课,起到承前启后的作用。通过数据结构课程的学习,使学生能使用数据结构的基本分析方法来提高编写程序的能力和应用计算机解决实际问题的能力。

数据结构一直是一门“教师难讲、学生难学”的课程,对于高职高专学生的教学,我们以“应用为主、够用就好”为原则,从实用出发,将数据结构中的知识进行取舍,精心编写本书。

为了让学生对计算机项目有个整体认识,本书以“数据结构实验系统”为主线来编写。每一章的主要算法构成一个相对独立的子系统,子系统在每章实验中给出,它既是授课重点,也是上机实验的主要内容。在最后一章中,将本书所有重要算法合并为一个大的实验系统。该系统设计有选择式菜单,可以通过选择菜单的各项来执行对应的算法,便于学生理解整个数据结构的知识,也可以由学生将书中未加入到该系统中的算法自行添到该系统中。

考虑到此课程的先导课是“C 语言程序设计”,书中算法均采用可在计算机上运行的 C 语言程序来描述。这样,降低了算法设计的难度,使学生能更直观形象地理解各算法。实现代码均在 VC6.0 中编译通过,并通过截屏给出了所有程序的运行结果。(截屏的好处是保证程序的正确性和结果的直观性。)

本教材共分 10 章。第 1 章概论,介绍了数据结构的基本概念和算法描述,并介绍了时间复杂度和空间复杂度;第 2 章~第 5 章,介绍了线性表、栈、队列和串等线性结构的逻辑特征、存储结构和常用算法的实现与基本应用;第 6 章~第 7 章,介绍了树和图两种非线性结构的逻辑特征、存储方法及相关算法的实现和基本应用;第 8 章查找,介绍了顺序查找、折半查找、分块查找和二叉排序树的查找方法及其实现;第 9 章排序,介绍了各种常用的排序方法及其实现;第 10 章数据结构实验系统开发及说明,主要介绍了如何将全书各章中实验程序合并成为一个完整的“数据结构实验系统”,并逐步进行调试直至最后运行。通过第 10 章的学习,学生可以建立起系统设计的初步概念。附录中提供了一套模拟试题和全书习题答案。

本书由刘畅主编,书中第 6 章~第 9 章由刘畅编写。汤晓伟、王茹、王象刚为副主编,书中第 1、第 2 章由汤晓伟编写,第 3、第 4 章由王茹编写,第 5、第 10 章由王象刚编写。叶宾、胡艳梅、张旭辉、郭畅老师编写了全书大量的插图、课后习题等内容。全书由刘畅统稿及修改,董凤服教授对本书的总体结构和写作方法提出很多宝贵意见并审阅了全书。

本书主要适用于高等职业院校的本、专科计算机专业学生,也适合于自学计算机相关知识的人员参考使用。本书讲授学时数为 60~70 学时,如果安排实训则将第 10 章的“数据结构实验系统”在 20 学时内完成。教师可根据学时数和学生的实际情况选讲本书的例子。

为配合本教学,本书还配套有相应电子课件,课件和书中所有算法及“数据结构实验系统”源代码可从电子工业出版社的华信教育资源网(<http://www.hxedu.com.cn>)下载,或通过邮件向刘畅老师索取,刘畅老师邮箱为: lcluwzc@163.com。

由于作者水平有限,书中疏漏或不足之处在所难免,恳请广大专家和读者给出宝贵意见。

编 者

目 录

第1章 概论	(1)
1.1 引 言	(1)
1.1.1 什么是数据结构	(1)
1.1.2 数据结构研究内容	(1)
1.2 数据的逻辑结构	(3)
1.2.1 有关概念和术语	(3)
1.2.2 数据的逻辑结构	(4)
1.3 数据的存储结构	(5)
1.4 算法和算法分析	(7)
1.4.1 算法的概念	(7)
1.4.2 算法的效率评价	(7)
小结	(9)
实验1 复习数组、指针、结构体	(10)
习题1	(13)
第2章 线性表	(16)
2.1 线性表的基本概念	(16)
2.1.1 线性表的定义	(16)
2.1.2 线性表的基本操作	(17)
2.2 线性表的顺序存储	(18)
2.2.1 顺序表	(18)
2.2.2 顺序表的基本运算的实现	(19)
2.3 线性表的链式存储	(25)
2.3.1 单链表	(26)
2.3.2 单链表的基本操作实现	(27)
2.3.3 循环链表	(32)
2.3.4 双向链表	(33)
2.4 顺序表与链表的比较	(35)
2.5 线性表的应用举例(约瑟夫环问题)	(36)
2.5.1 问题描述	(36)
2.5.2 数据结构	(36)
2.5.3 程序流程	(36)
2.5.4 源程序及运算结果	(37)
小结	(40)
实验2 线性表子系统	(40)
习题2	(46)
第3章 栈	(49)
3.1 栈的定义及运算	(49)
3.1.1 栈的定义	(49)

3.1.2 栈的基本操作	(50)
3.2 栈的存储和实现	(50)
3.2.1 顺序栈	(50)
3.2.2 链栈	(55)
3.3 栈的应用举例	(58)
小结	(61)
实验3 栈子系统	(61)
习题3	(69)
第4章 队列	(72)
4.1 队列的定义与基本运算	(72)
4.1.1 队列 (Queue) 的定义	(72)
4.1.2 队列的基本操作	(73)
4.2 队列的存储实现及运算实现	(73)
4.2.1 顺序队列	(73)
4.2.2 循环队列	(74)
4.2.3 链队列	(79)
4.3 队列的应用举例	(82)
小结	(83)
实验4 队列子系统	(83)
习题4	(87)
第5章 串和广义表	(91)
5.1 串的定义和基本运算	(91)
5.1.1 串的定义	(91)
5.1.2 串的基本操作	(92)
5.2 串的存储结构	(93)
5.2.1 定长顺序串存储	(94)
5.2.2 链式存储	(95)
5.2.3 串的堆分配存储结构	(95)
5.3 串的基本运算的实现	(97)
5.4 广义表	(101)
5.4.1 广义表的定义和运算	(101)
5.4.2 广义表的存储	(102)
小结	(104)
实验5 串子系统	(105)
习题5	(113)
第6章 树	(117)
6.1 树	(117)
6.1.1 树的定义	(117)
6.1.2 树的基本术语	(118)
6.2 二叉树	(119)
6.2.1 二叉树的定义	(119)
6.2.2 二叉树的基本运算	(120)

6.2.3	二叉树的性质	(120)
6.2.4	二叉树的存储结构	(121)
6.3	遍历二叉树	(123)
6.3.1	遍历二叉树	(123)
6.3.2	恢复二叉树	(125)
6.3.3	二叉树的其他操作	(127)
6.4	树和森林与二叉树的转换	(129)
6.4.1	树的存储结构	(129)
6.4.2	树和森林转换为二叉树	(131)
6.4.3	二叉树还原为树和森林	(133)
6.5	哈夫曼树	(135)
6.5.1	相关概念和哈夫曼树的定义	(135)
6.5.2	哈夫曼树的构造方法	(136)
6.5.3	哈夫曼编码	(137)
	小结	(140)
	实验 6 二叉树子系统	(141)
	习题 6	(147)
第 7 章	图	(151)
7.1	图的定义和术语	(151)
7.1.1	图的定义	(151)
7.1.2	图的相关术语	(152)
7.1.3	图的基本操作	(154)
7.2	图的存储结构	(154)
7.2.1	邻接矩阵	(155)
7.2.2	邻接表	(157)
7.3	图的遍历	(160)
7.3.1	深度优先搜索法	(161)
7.3.2	广度优先搜索法	(162)
7.4	图的连通性	(164)
7.4.1	无向图的连通分量和生成树	(164)
7.4.2	最小生成树的基本概念	(165)
7.4.3	普里姆 (Prim) 算法	(165)
7.4.4	克鲁斯卡尔 (Kruskal) 算法	(169)
7.5	最短路径问题	(171)
7.5.1	最短路径概念	(171)
7.5.2	求单源最短路径的方法	(172)
	小结	(176)
	实验 7 图子系统	(176)
	习题 7	(181)
第 8 章	查找	(185)
8.1	基本概念	(185)
8.2	静态查找表	(186)

8.2.1 顺序查找	(187)
8.2.2 折半查找	(188)
8.2.3 分块查找	(191)
8.3 动态查找表	(193)
8.3.1 二叉排序树的概念	(193)
8.3.2 二叉排序树的基本运算	(194)
8.4 哈希表查找	(200)
8.4.1 哈希表查找的基本思想	(200)
8.4.2 哈希表的构造方法	(201)
8.4.3 哈希表的冲突处理	(202)
8.4.4 哈希表的查找及性能分析	(204)
小结	(206)
实验 8 查找子系统	(207)
习题 8	(216)
第 9 章 排序	(219)
9.1 排序的基本概念	(219)
9.2 插入排序	(220)
9.2.1 直接插入排序	(220)
9.2.2 希尔排序 (Shell's Sort)	(222)
9.3 交换排序	(224)
9.3.1 冒泡排序 (Bubble Sort)	(224)
9.3.2 快速排序 (Quick Sort)	(225)
9.4 选择排序	(227)
9.4.1 直接选择排序	(227)
9.4.2 堆排序 (Heap Sort)	(229)
9.5 归并排序	(232)
9.6 各种排序方法的比较	(234)
小结	(234)
实验 9 排序子系统	(235)
习题 9	(241)
第 10 章 数据结构实验系统开发及说明	(245)
10.1 系统设计	(245)
10.1.1 系统模块设计规则	(245)
10.1.2 系统中的文件包含	(246)
10.2 系统实现	(247)
10.2.1 主调函数的设计与实现	(247)
10.2.2 调试运行系统	(248)
10.2.3 实训报告	(250)
附录 A 习题答案	(251)
附录 B 模拟试题及答案	(270)
参考文献	(276)

第1章 概 论

数据作为计算机加工处理的对象，在计算机中如何表示、存储是计算机科学研究的主要内容之一，更是计算机技术需要解决的问题之一。要使计算机能够更有效地进行这些非数值性处理，就必须弄清楚这些操作对象的特点，在计算机中的表示方式以及各个操作的具体实现手段。这些就是《数据结构》这门课程研究的主要内容。



本章学习目标

- 数据结构研究的主要内容；
- 数据结构中涉及的相关概念；
- 算法的概念、描述方法以及评价标准。

1.1 引 言

1.1.1 什么是数据结构

“数据结构”作为计算机科学技术专业的专业基础课，是十分重要的核心课程，其主要的研究内容就是数据之间的逻辑关系和物理实现，探索有利的数据组织形式及存取方式。所有计算机系统软件和应用软件的设计、开发都要用到各种类型的数据结构。因此，要想更好地运用计算机来解决实际问题，仅掌握几种计算机程序设计语言是难以应付众多复杂项目的。要想有效地使用计算机、充分发挥计算机的性能，还必须学习和掌握数据结构的有关知识。

数据结构包含两方面的内容，其一是构成集合的数据元素，其二是数据元素之间存在的关系。数据结构也就是带有结构的数据元素的集合，结构指的是数据元素之间的相互关系，即数据的组织形式。由此可见，计算机所处理的数据是具有内在联系的集合。著名的计算机科学家 N. Wirth 提出“算法+数据结构程序”的思想，明确地指出了数据结构实际上是程序的主要部分。

数据结构是介于数学、计算机硬件和计算机软件三者之间的一门核心课程。在计算机科学中，数据结构不仅是一般非数值计算机程序设计的基础，还是设计和实现汇编语言、编译程序、操作系统、数据库系统，以及其他系统程序和大型应用程序的重要基础。

从我国的计算机教学现状来看，数据结构不仅是计算机专业教学计算机中的核心课程之一，而且已经逐步成为非计算机专业学生的重要选修课程。

1.1.2 数据结构研究内容

用计算机解决一个实际问题时，大致需要以下几个步骤：

- (1) 从具体问题抽象出一个适当的数学模型；

- (2) 设计求解数学模型的算法;
- (3) 编制、运行并调试程序,直到解决实际问题。

所以,建立数学模型的过程就是分析问题,并找出这些对象的关系,用数学语言来描述。下面通过几个实例来了解一下如何构造数据结构。

【例 1-1】 学生的学籍档案管理表。

假设一个学籍档案管理系统应包含如表 1-1 所示的学生信息。

表 1-1 学生信息登记表

学 号	姓 名	性 别	出生年月日	入 学 总 分	
20110101	王 新	男	1992 年 11 月 3 日	426	
20110102	刘智勇	男	1992 年 1 月 24 日	451	
20110103	赵婷婷	女	1991 年 12 月 3 日	477	
20110104	张鹏飞	男	1992 年 6 月 18 日	439	
.....					

将表 1-1 称为一个数据结构,表中的每一行是一个结点,或称为记录 (Record),它由学号、姓名、性别、出生年月日、入学总分等数据项 (Item) 组成。在这个表中,第一条记录没有直接前驱,称为开始结点;最后一条记录没有直接后继,称为终端结点。除了第一条记录以外,其余的记录都有且只有一条直接前驱记录和一条直接后继记录。这些结点之间的关系是“一对一”的关系,这就是该表的逻辑结构。

此外,该表在计算机的外存中是如何存储的?表中各元素是存储在连续的单元中(顺序结构),还是用指针链接存储在不连续的存储单元(链式存储)?这就构成了该表的存储结构。

对该表进行的查找记录、插入记录、删除记录、对记录进行排序、统计等,就构成了数据的运算。

数据结构就是研究数据的逻辑、存储结构和运算方法(即算法)的学科。

【例 1-2】 井字棋人机对弈问题。

井字棋对弈过程中,任何一方只要使相同的三个棋子连成一条直线即为胜方(可以是一行、一列或一条对角线)。如果下一棋子由“x”方下,可以派生出五个子格局,如图 1-1 所示。

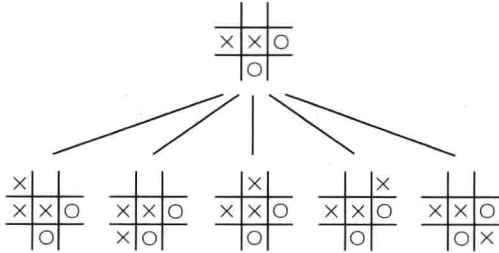


图 1-1 人机对弈图

若将从对弈开始到结束的过程中所有的可能的格局画在一张图上,即形成了一棵倒挂的对弈“树”。“树根”是对弈开始时的第一步棋,而所有的“叶子”便是可能出现的结局。在本例中,对弈开始之前的棋盘格局没有直接前驱,称为开始结点(即根),以后每走一步

棋，都有多种对应的策略，结点之间存在着“一对多”的关系，它构成了井字棋对弈的逻辑结构。

【例 1-3】 制订教学计划。

在制订教学计划时，需要考虑各门课程的开设顺序。有些课程需要先修课程，有些课程则不需要，而有些课程又是其他课程的先修课程。比如，计算机专业课程的开设情况如表 1-2 所示。

表 1-2 计算机专业学生的必修课程

课 程 编 号	课 程 名 称	先 修 课 程
C ₁	高等数学	无
C ₂	程序设计基础	无
C ₃	普通物理	C ₁
C ₄	离散数学	C ₁ ,C ₂
C ₅	C 语言程序设计	C ₂
C ₆	数据结构	C ₂ ,C ₄ ,C ₅
C ₇	编译原理	C ₅ ,C ₆
C ₈	计算机原理	C ₃
C ₉	操作系统	C ₆ ,C ₈

教学计划的关系图如图 1-2 所示。

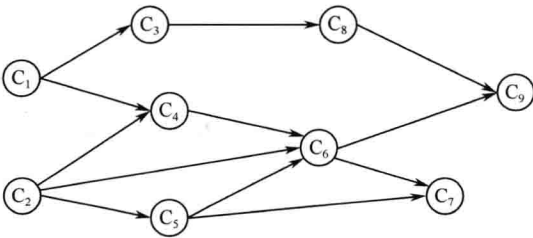


图 1-2 教学计划的关系图

在图 1-2 中，各结点都有多个直接前驱和多个直接后继，结构之间存在着“多对多”的关系，称为图。

综上实例可知，非数值计算问题的数学模型不再是数学方程的问题，而是诸如上述的表、树、图之类的数据结构。因此，数据结构是一门研究非数值计算的程序设计问题的学科，主要研究数据的逻辑结构、存储结构和算法。

下面介绍一下这三方面的知识。

1.2 数据的逻辑结构

1.2.1 有关概念和术语

在系统地学习数据结构知识之前，先了解一些相关概念和术语。

1. 数据 (Data)

数据是指所有能输入到计算机中并被计算机程序处理的符号的表示。在计算机中,所谓数据就是计算机加工处理的对象,它可以是数值数据,也可以是非数值数据。数值数据包括整数、实数等,主要用于科学计算、财会和商务处理等;非数值数据则包括文字、符号、图形、动画、声音等。

2. 数据元素 (Data Element)

数据元素(也称为结点)是数据的基本单位,在计算机程序中通常作为一个整体进行考虑和处理。一个数据元素可以由若干个数据项组成。例如,一个学生的信息为一个数据元素,而一个学生的信息中的每一项(如学号、姓名、性别等)为一个数据项。数据项是数据处理中不可分割的最小单位。

3. 数据结构 (Data Structure)

按某种逻辑关系组织起来的一批数据,按一定的映像方式把它存放在计算机存储器中,并在这些数据上定义了一个运算的集合,就叫做数据结构。

数据结构一般包括以下 3 个方面内容:

- (1) 数据元素之间的逻辑关系,也称数据的逻辑结构;
- (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;
- (3) 数据的运算,即对数据施加的操作。

1.2.2 数据的逻辑结构

1. 定义

数据的逻辑结构是指数据元素之间逻辑关系描述。可以用一个二元组表示,其形式化描述为:

$$G=(D, R)$$

其中, D 为数据元素的有限集合, R 为 D 上关系的有限集合。数据的逻辑结构是从逻辑关系上描述数据,与数据的存储无关,是独立于计算机的。

2. 数据的逻辑结构的分类

根据数据元素之间的逻辑关系的不同特性,分为下列 4 类基本结构,如图 1-3 所示。

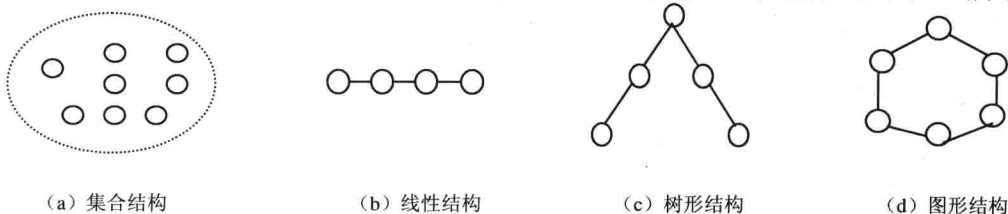


图 1-3 数据结构的 4 种基本逻辑结构

(1) 集合。结构中的数据元素之间除了“同属于一个集合”的关系外,别无其他关系,这是一种最简单的数据结构。

(2) 线性结构。结构中的数据元素之间存在着“一对一”的关系。

线性结构的特点：表中数据元素之间是一种先后关系，对于表中任一结点，与它相邻且在它前面的结点（称为直接前驱）最多只有一个；与表中任一结点相邻且在其后的结点（称为直接后继）也最多只有一个。这种关系称为“线性结构”。

(3) 树形结构。结构中的数据元素之间存在着“一对多”的关系。

树形结构的特点：数据元素之间是一对多关系，即一个数据元素向上和一个数据元素相连（称为双亲结点），向下和多个数据元素相连（称为孩子结点）。这种关系称为“树形结构”。

(4) 图形结构或网状结构。结构中的任意数据元素之间都可以有关系，元素之间存在着“多对多”的关系。

图形结构的特点：图中数据元素存在着多对多的任意关系。一个结点可能有多个直接前驱和直接后继。

1.3 数据的存储结构

数据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。数据的存储结构是逻辑结构在计算机存储器中的实现。本书将介绍常用的两种基本的存储结构：顺序存储结构和链式存储结构。

数据的逻辑结构和存储结构的关系是：存储结构是逻辑关系的映像与元素本身映像，是数据结构的实现；逻辑结构是数据结构的抽象。

1. 顺序存储结构

顺序存储结构：借助元素在存储器中的相对位置来表示数据元素间的逻辑关系。

【例 1-4】 对于表 1-1 提出的学生信息登记表进行存储，假定每个元素占用 50 个存储单元，数据从 1000 号单元开始由低地址向高地址存放，对应的顺序存储结构如表 1-3 所示。

表 1-3 表 1-1 的学生信息登记表的顺序存储表

存储地址	学号	姓名	性别	出生年月日	入学总分	
1000	20110101	王新	男	1992 年 11 月 3 日	426	
1050	20110102	刘智勇	男	1992 年 1 月 24 日	451	
1100	20110103	赵婷婷	女	1991 年 12 月 3 日	477	
1150	20110104	张鹏飞	男	1992 年 6 月 18 日	439	
.....						

在 C 语言中，最简单的方法是用一维数组来实现顺序存储。

顺序存储结构的主要特点如下。

- ◆ 可实现对各数据元素的随机访问。这是因为只要知道存储的首地址以及每个数据元素所占的存储单元，就可以计算出各数据元素的存储地址。
- ◆ 不利于修改，在对数据元素进行插入、删除运算时可能要移动一系列的数据元素。

2. 链式存储结构

链式存储结构特点是借助指示元素存储地址的指针表示数据元素间的逻辑关系。

【例 1-5】 对于表 1-1 学生信息登记表进行链式存储时，在每个数据元素后方附加一个指向“下一个结点地址”的指针字段，用于存放后继数据元素的存储地址，每个数据元素的地址是随机的，可以不连续。对应的链式存储结构见表 1-4 所示。

表 1-4 表 1-1 学生信息登记表的链式存储表

存 储 地 址	学 号	姓 名	性 别	出生年月日	入 学 总 分	下一结点地址	
2320	20110101	王 新	男	1992 年 11 月 3 日	426	2670	
2450	20110102	刘智勇	男	1992 年 1 月 24 日	451	3690	
2670	20110103	赵婷婷	女	1991 年 12 月 3 日	477	2450	
3690	20110104	张鹏飞	男	1992 年 6 月 18 日	439	4230	
.....							

链式存储结构的主要特点：

- ◆ 利于修改，在对数据元素进行插入、删除运算时，仅需修改数据元素的指针字段值，而不必移动数据元素；
- ◆ 由于逻辑上相邻的数据元素在存储位置中不一定相邻，因此，链式存储结构不能对数据元素进行随机访问。

3. 索引存储结构

索引存储是在原有的存储结构的基础上，附加建立一个索引表，索引表中的每一项都由关键字和地址组成。索引表反映了按某一个关键字递增或递减排列的逻辑次序。采取索引存储的主要作用是为了提高数据的检索速度。

4. 散列存储结构

散列存储是通过构造散列函数来确定数据存储地址或查找地址。例如，某一地区进行 2000 年以后出生的人口统计。我们用“出生年份-2000=存储地址”来构造一个函数，即用出生年份减去 2000 得到的差值就是存储地址，这样就能方便的得到这样的调查表，如表 1-5 所示。

表 1-5 2000 年后出生人口调查表

存 储 地 址	出 生 年 份	人 数	存 储 地 址	出 生 年 份	人 数
01	2001	687	07	2007	762
02	2002	782	08	2008	711
03	2003	805	09	2009	658
.....			10	2010	690

1.4 算法和算法分析

1.4.1 算法的概念

1. 算法的定义

算法是由若干条指令所组成的有穷序列，其中每条指令表示计算机的一个或多个操作。

2. 算法的特性

一个好的算法应该具有以下 5 个特性。

(1) 有穷性。一个算法必须（对任何合法的输入值）在执行有限时间内完成，不能形成无穷循环。

(2) 确定性。算法中的每一条指令必须有确切的含义，不会产生二义性。

(3) 可行性。算法中描述的操作都可以通过已经实现的基本运算执行有限次来实现。

(4) 输入。一个算法有零个或多个输入，这些输入取自于某个特定对象的集合。

(5) 输出。一个算法必有一个或多个输出，这些输出是与输入有着一定关系的量。

3. 好算法应达到的目标

算法与数据结构是相辅相成的。解决某一特定类型问题的算法可以选定不同的数据结构，而且选择恰当与否直接影响算法的效率。反之，一种数据结构的优劣由各种算法的执行来体现。

设计一个好的算法通常需要考虑以下几个方面的要求。

(1) 正确性。要求算法能够正确地执行预先规定的功能，并达到所期望的性能要求。

(2) 可读性。为了便于理解、测试和修改算法，算法应该具有良好的可读性。

(3) 健壮性。当输入非法的数据时，算法应能恰当地做出反应或进行相应处理，而不是产生莫名其妙的输出结果。并且处理出错的方法不应是中断程序的执行，而是返回一个表示错误或错误性质的值，以便在更高的抽象层次上进行处理。

(4) 高效性。要求算法的执行时间要尽可能的短，算法的效率就越高。

(5) 低存储量。完成相同的功能，执行算法时所占用的附加存储空间要尽可能的少。

实际上，算法很难做到十全十美，因为上述要求有时会相互抵触。所以，实际操作中，应以算法正确为前提，根据具体情况而有所侧重。

1.4.2 算法的效率评价

一个好的算法首先要具备正确性、可读性和健壮性。在具备了这 3 个条件后，就应考虑算法的效率问题，即算法的时间效率（所需运算的时间）和空间效率（所占存储空间）两方面。对于求解同一个问题，可以设计出若干个算法，对于不同的算法进行性能分析是数据结构的一个重要内容。在算法满足正确性的前提下，如何评价不同算法的优劣呢？通常主要考虑算法的时间复杂度和空间复杂度这两方面。一般情况下，鉴于运算空间（内存）较为充足，所以把算法的时间复杂度作为重点分析。

1. 时间复杂度 (Time Complexity)

一个算法所需的运算时间通常与所解决问题的规模大小有关。问题规模是一个和输入有关的量,用 n 表示问题规模的量,通常把算法运行所需的时间 T 表示为 n 的函数,记为 $T(n)$ 。不同的 $T(n)$ 算法,当 n 增长时,运算时间增长的快慢很不相同。一个算法所需的执行时间就是该算法中所有语句执行次数之和。当 n 逐渐增大时 $T(n)$ 的极限情况,一般简称为时间复杂度。

当讨论一个程序的运行时间时,注重的不是 $T(n)$ 的具体值,而是它的增长率。 $T(n)$ 的增长率与算法中数据的输入规模紧密相关,而数据输入规模往往用算法中的某个变量的函数来表示,通常是 $f(n)$ 。随着数据输入规模的增大, $f(n)$ 的增长率与 $T(n)$ 的增长率相近,因此 $T(n)$ 同 $f(n)$ 在数量级上是一致的。记作:

$$T(n) = O(f(n))$$

其中,大写字母 O 表示数量级 (Order of Magnitude) 的概念, $f(n)$ 为函数形式。例如,若 $T(n) = 3n^2 + 5n + 2$,则 $3n^2 + 5n + 2$ 的数量级与 n^2 的数量级相同,所以 $T(n) = O(n^2)$ 。

注意,当 $T(n)$ 为多项式时,可只取其最高次幂项并省略其系数,其他的次幂项及系数均略去不写。一般地,对于足够大的 n ,常用的时间复杂度 (如图 1-4 所示) 的大小次序如下:

$$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < O(2^n)$$

算法时间复杂度的数量级越大,表示该算法的效率越低,反之越高。例如, $O(1)$ 为常数数量级,即算法的时间复杂性与输入规模 n 无关。

【例 1-6】 分析以下算法的时间复杂度。

```
1  x=0;y=0;
2  for(k=1; k<=n; k++)
3      x++;                                (1) 执行 n 次
4  for(i=1; i<=n; i++)
5      for(j=1; j<=n; j++)
6          y++;                            (2) 执行 n^2 次
```

解: 本程序有两个循环体,第一个循环的时间复杂度为 n ,第二个循环是双重循环,其时间复杂度为 n^2 ,所以上面程序段的时间复杂度为:

$$T(n) = n + n^2$$

则其时间复杂度为:

$$T(n) = O(n^2)$$

【例 1-7】 分析以下算法的时间复杂度。

```
1  i=1;
2  while(i<=n)
3      i=2*i;                                (1) 执行 f(n) 次
```

解: 设第 3 行语句 (1) 执行次数是 $f(n)$,则 $2^{f(n)} \leq n$,所以 $T(n) = O(\log_2 n)$ 。

【例 1-8】 求两个矩阵相乘的函数的时间复杂度。

```
1  void mult(int a[][N], int b[][N], int c[][N])
2  { /*以二维数组存储矩阵元素, c 为 a 和 b 的乘积*/
3      for(i=0; i<n; i++)                    (1) 执行 n 次
4          for(j=0; j<n; j++)                (2) 执行 n^2 次
5              { c[i][j]=0;
6                  for(k=0; k<n; k++)        (3) 执行 n^3 次
```

```

7          c[i][j] += a[i][k] * b[k][j];
8      }
9  }

```

解: 嵌套循环为每层循环次数的乘积, 因为该函数为三重循环, 所以时间复杂度为 $O(n^3)$ 。

2. 空间复杂度 (Space Complexity)

一个算法的空间复杂度是指程序运行开始到结束所需要的存储空间。包括算法本身所占用的存储空间、输入/输出数据占用的存储空间以及算法在运行过程中的工作单元和实现算法所需辅助空间。类似于算法的时间复杂度, 算法所需存储空间的量度记作:

$$S(n) = O(f(n))$$

其中, n 为问题的规模, 一个程序上机执行时, 除了需要存储空间来存放本身所用的指令、常数、变量和输入数据以外, 还需要一些对数据进行操作的工作单元和实现算法所必需的辅助空间。在进行时间复杂度分析时, 如果所占空间量依赖于特定的输入, 一般都是按最坏情况分析。

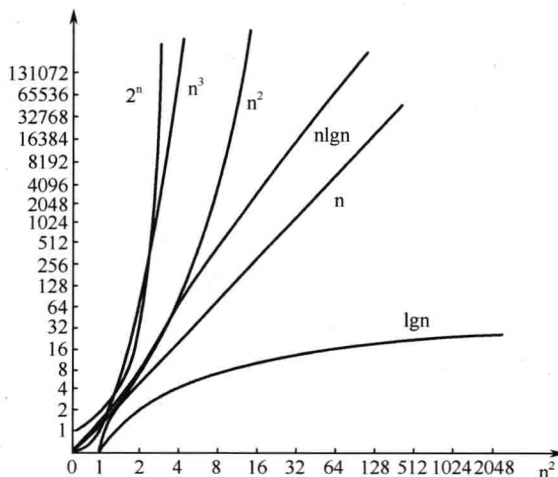


图 1-4 常见函数的增长率

小 结

本章主要介绍了有关数据结构的以下几方面。

- (1) 数据结构主要研究数据的逻辑结构、存储结构和运算方法。
- (2) 数据的逻辑结构包括: 集合、线性结构、树形结构、图形结构 4 种基本类型。
- (3) 数据的存储结构包括: 顺序存储结构、链式存储结构、索引存储、散列存储 4 种。
- (4) 顺序存储可以采用一维数组来存储; 链式存储可以采用链表来存储; 索引存储则在原有存储数据结构的基础上, 附加建立一个索引表来实现, 主要作用是为了提高数据的检索速度; 而散列存储则是通过构造散列函数来确定数据存储地址或查找地址。
- (5) 算法是对特定问题求解步骤的一种描述, 是指令的有限序列。算法具有: 有穷性、确定性、正确性、输入、输出等特性。
- (6) 一个好的算法应达到: 正确性、可读性、健壮性、高效性和低存储量等目标。