

高等学校计算机科学与技术教材

# 数据结构基础教程

*Computer*

文益民 主编

文益民 郭杰 李健 编著

周学毛 主审

. 12



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社

<http://press.bjtu.edu.cn>

高等学校计算机科学与技术教材

# 数据结构基础教程

文益民 主编

文益民 郭杰 李健 编著

周学毛 主审

清华大学出版社

北京交通大学出版社

·北京·

## 内 容 简 介

本书系统地介绍了各种常用的数据结构及排序、查找的各种算法,阐述了各种数据结构的逻辑结构、存储结构及其基本运算。各数据结构类型和基本运算,首先用类 C 代码描述,然后用可编译运行的 C 语言代码实现,并给出了详细的注释。全书既注重原理又强调实践,配有大量的图表和习题,概念讲解清楚、逻辑性强、可读性好。本书的特点在于,首次尝试在基础课程中介绍计算机科学发展史知识,采用脚注的形式使学生了解计算机科学史知识和数据结构课程与其他课程之间的关系;附有大量以“思考”形式出现的问题,以便在恰当的时机引导学生思考,启发思维;以学生为主体精心设计了数据结构课程的实践教学内容。

本书可作为高等院校非计算机专业教材或高职、高专院校计算机专业教材,也可作为成人教育(面授或函授)的教材,还可为参加全国计算机软件水平程序员等级考试提供参考,亦可供广大从事计算机应用的科技人员参考。

**版权所有,翻印必究。**

**本书封面贴有清华大学出版社防伪标签,无标签者不得销售。**

(本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。)

### 图书在版编目(CIP)数据

数据结构基础教程 / 文益民主编;文益民,郭杰,李健编著. —北京:清华大学出版社;北京交通大学出版社, 2005.1

(高等学校计算机科学与技术教材)

ISBN 7-81082-456-2

I. 数… II. ①文… ②文… ③郭… ④李… III. 数据结构-高等学校-教材 IV. TP311.12

中国版本图书馆 CIP 数据核字(2004)第 122641 号

责任编辑:韩 乐 特邀编辑:朱 宇

出版者:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686414

印刷者:北京瑞达方舟印务有限公司

发行者:新华书店总店北京发行所

开 本:185×260 印张:14 字数:349千字

版 次:2005年1月第1版 2005年1月第1次印刷

书 号:ISBN 7-81082-456-2/TP·164

印 数:1~5 000册 定价:21.00元

# 前 言

1997年5月,一台名为“深蓝”的超级计算机将棋盘上的一个兵走到C4位置时,人类有史以来最伟大的国际象棋名家卡斯帕罗夫不得不沮丧地服输,上世纪末的一场人机大战终于以计算机的微弱优势取胜。2003年4月14日人类基因组计划宣告完成,后基因组时代已经到来,各种生物学成果在计算中不断出现。这些成果使得人们再一次认识到计算的重要性。如今,计算技术已广泛地应用到各个领域,后因特网时代已经到来,以信息化带动工业化已成为时代的主题。计算机程序设计是计算技术的重要内容,而数据结构是计算机程序设计的重要理论基础,它不仅是计算机学科的核心课程,而且已成为其他专业的热门选修课。数据结构课程的教学目标是学会分析需要计算机处理的数据对象的特性,以选择适当的数据结构和存储结构,从而选择相应的算法;初步掌握算法性能分析的方法。通过本课程的学习,使学生获得更进一步的程序设计技能训练,提高编程能力。

长久以来,由于数据结构课程自身的抽象性和严密性,以及数据结构开设的时间通常在刚入大学的第二学期,教师大都感觉数据结构课程难教,学生普遍反映数据结构课程难学,学生很难独立完成算法的实现。基于上述问题,我们在编写本教材时充分考虑了学生的知识结构和教师的教学方法。本教材的编写宗旨是,既注重原理又注重实践,既注重抽象思维又注重形象思维,既方便自学又方便教学。本书的编写具有以下特色。

1. 首次尝试在计算机专业的核心基础课程中增加计算机科学知识,使学生在对本教材的学习过程中,不但能学到专业知识,还能了解计算机科学发展历史的相关知识和数据结构课程与其他课程的联系。对提高学生对本课程的兴趣,拓宽学生的知识面大有益处。

2. 在教材中使用“▲思考”标志,提出问题拓展学生思维。思考不同于习题,习题的作用代替不了思考。因为习题在每个章节之后,一般要等讲完一个章节才会遇到,因此习题对于课堂教学是滞后的。采用提示思考的方式,可以在教学中恰到好处地启发学生的思维。教材使用“▲思考”标志通常有三种情况:一种是提醒学生需要注意的内容,一种是启发学生基于当前知识基础进一步思考的内容,第三种是提示本教材没有讲授的内容,以引导学有余力的学生拓展自身的知识面。

3. 以学生为主体设计数据结构课程的实践内容。数据结构课程是计算机专业的一门重要的基础课程,其性质是实践性的。如何引导学生利用所学概念和算法进行实践是这门课程成败的关键,为此我们采取了两个措施:一是注重概念和算法的应用背景,对每种数据结构都有应用实例,有代码实现;二是在基本运算的实现代码中留有要求学生自己实现代码的部分。这就要求学生不但要读懂已经实现的部分代码,还要自己设计代码,最后才能得到一个完整实现的系统。这有利于提高学生对数据结构概念和算法的理解,真正提高学生的编程能力。每章后的上机实习就由这两部分内容组成,本书中提供的实现代码均在VC 6.0中编译通过。

本书介绍了各种最常用的数据结构,阐述了各种数据结构内涵的逻辑关系,讨论了它们在计算机中的存储表示,以及基于这些数据结构的运算和实际的执行算法,并对算法的效率进行了简要的分析和讨论。

全书包含 9 章内容, 第 1 章介绍了数据结构和算法分析的基本概念; 第 2 章到第 5 章第 1 节介绍了线性结构; 第 5 章第 2 节到第 7 章介绍了非线性结构, 其中有多维数组、广义表、树和图; 第 8 章和第 9 章分别介绍了排序和查找。为了便于讲授, 所有数据类型定义和算法实现首先均用类 C 语言 (或 C 语言) 描述。为了便于学生上机实习, 除了留给学生实习的内容外, 所有基本运算算法均用 C 语言实现, 学生可以直接编译运行。另外, 为了避免 C 语言中数组的第一个元素的下标是 0 给学习和讲授带来的不便, 本书在没有特别申明的地方均不使用 C 语言中数组下标为 0 的元素。

本书可作为本科院校非计算机专业教材或高职、高专计算机专业的教材, 讲授 60 课时左右。本书的每一章末均有上机实习和习题。根据学生的实际情况, 每章的上机实习可安排 2 课时左右, 这样总计实验课时约为 16 个小时。本书还可作为成人教育 (面授或函授) 的教材, 也可为参加全国计算机软件水平程序员等级考试提供参考, 亦可供从事计算机应用的科技人员参考。

本书由文益民整体构思, 在 3 位多年从教计算机专业数据结构课程作者的教学经验基础上, 经过多次反复磋商和共同讨论定稿, 是这 3 位作者共同合作的产物。本书经湖南大学计算机与通信学院周学毛副教授详细审阅, 提出了许多宝贵意见。上海交通大学仿脑计算实验室的硕士、博士生们提供了许多有益的建议, 作者谨此一并致以诚挚的谢意。

本书的第 1、2、4、5 和第 7 章由文益民编写, 第 3 章和第 6 章由郭杰编写, 第 8 章和第 9 章由李健编写。全书由文益民统稿、修改。

由于编著者水平有限, 书中疏漏或错误在所难免, 殷切期望广大读者批评指正。

文益民 郭杰 李健

2004 年 12 月

# 目 录

第 1 章 绪论 .....	(1)
1.1 数据结构基本概念 .....	(1)
1.1.1 数据结构实例 .....	(1)
1.1.2 数据结构概念 .....	(3)
1.2 算法分析基本概念 .....	(5)
1.2.1 算法 .....	(5)
1.2.2 算法效率分析 .....	(5)
1.2.3 算法效率评价 .....	(6)
习题 1 .....	(7)
第 2 章 线性表 .....	(10)
2.1 概念和运算 .....	(10)
2.1.1 线性表概念 .....	(10)
2.1.2 线性表基本运算 .....	(10)
2.2 顺序存储结构 .....	(11)
2.2.1 顺序表 .....	(11)
2.2.2 顺序表基本运算 .....	(12)
2.3 链式存储结构 .....	(15)
2.3.1 线性链表 .....	(15)
2.3.2 线性链表基本运算 .....	(17)
2.4 线性表应用 .....	(20)
2.5 基本运算实现 .....	(22)
2.5.1 顺序表基本运算实现 .....	(22)
2.5.2 链表基本运算实现 .....	(26)
上机实习 线性表 .....	(30)
习题 2 .....	(30)
第 3 章 栈 .....	(33)
3.1 概念和运算 .....	(33)
3.1.1 栈概念 .....	(33)
3.1.2 栈基本运算 .....	(33)
3.2 存储和实现 .....	(34)
3.2.1 顺序栈 .....	(34)
3.2.2 链栈 .....	(36)
3.3 栈应用 .....	(37)
3.3.1 数制转换 .....	(37)
3.3.2 表达式求值 .....	(39)

3.3.3 栈和递归 .....	(42)
3.4 栈基本运算实现 .....	(44)
3.4.1 顺序栈基本运算实现 .....	(44)
3.4.2 链栈基本运算实现 .....	(47)
上机实习 栈 .....	(49)
习题 3 .....	(50)
<b>第 4 章 队列 .....</b>	<b>(52)</b>
4.1 概念和基本运算 .....	(52)
4.1.1 队列概念 .....	(52)
4.1.2 队列基本运算 .....	(52)
4.2 顺序存储结构和运算 .....	(53)
4.3 循环队列 .....	(54)
4.4 链队列 .....	(56)
4.5 队列应用 .....	(58)
4.6 队列基本运算实现 .....	(60)
4.6.1 循环队列运算实现 .....	(60)
4.6.2 链队列运算实现 .....	(63)
上机实习 队列 .....	(65)
习题 4 .....	(66)
<b>第 5 章 线性结构推广 .....</b>	<b>(68)</b>
5.1 串 .....	(68)
5.1.1 定义 .....	(68)
5.1.2 基本运算 .....	(68)
5.1.3 定长顺序存储 .....	(70)
5.1.4 模式匹配 .....	(72)
5.1.5 链式存储结构 .....	(74)
5.2 数组 .....	(75)
5.2.1 定义和存储 .....	(75)
5.2.2 矩阵压缩存储 .....	(76)
5.3 广义表 .....	(79)
5.3.1 定义 .....	(79)
5.3.2 存储 .....	(80)
5.4 串的基本运算实现 .....	(81)
上机实习 串 .....	(86)
习题 5 .....	(87)
<b>第 6 章 树 .....</b>	<b>(89)</b>
6.1 树的概念和基本运算 .....	(89)
6.1.1 定义 .....	(89)
6.1.2 基本术语 .....	(90)

6.1.3 基本运算 .....	(91)
6.2 树的存储 .....	(91)
6.3 二叉树的概念和性质 .....	(94)
6.3.1 概念和基本运算 .....	(94)
6.3.2 性质 .....	(94)
6.3.3 存储 .....	(96)
6.4 二叉树遍历 .....	(97)
6.5 线索二叉树 .....	(100)
6.6 树和二叉树 .....	(101)
6.6.1 树与二叉树的转换 .....	(101)
6.6.2 二叉树与森林的转换 .....	(102)
6.7 哈夫曼树及其应用 .....	(104)
6.8 二叉树基本运算 .....	(106)
6.9 二叉树基本运算实现 .....	(108)
上机实习 二叉树 .....	(112)
习题 6 .....	(112)
<b>第 7 章 图 .....</b>	<b>(115)</b>
7.1 概念和基本运算 .....	(115)
7.1.1 图的概念 .....	(115)
7.1.2 图的基本运算 .....	(118)
7.2 图存储 .....	(118)
7.2.1 数组表示法 .....	(119)
7.2.2 邻接表 .....	(121)
7.3 图遍历 .....	(123)
7.3.1 连通图的深度优先搜索遍历 .....	(123)
7.3.2 广度优先搜索 .....	(125)
7.4 最小生成树 .....	(126)
7.4.1 Prim 算法 .....	(127)
7.4.2 Kruskal 算法 .....	(129)
7.5 单源点最短路径 .....	(130)
7.6 图的基本运算实现 .....	(133)
上机实习 图 .....	(140)
习题 7 .....	(140)
<b>第 8 章 排序 .....</b>	<b>(142)</b>
8.1 排序基本概念 .....	(142)
8.2 插入类排序 .....	(143)
8.2.1 直接插入排序 .....	(143)
8.2.2 折半插入排序 .....	(145)
8.2.3 希尔排序 .....	(146)



8.3 交换类排序 .....	(148)
8.3.1 冒泡排序 .....	(148)
8.3.2 快速排序 .....	(150)
8.4 选择类排序 .....	(153)
8.4.1 简单选择排序 .....	(153)
8.4.2 树型选择排序 .....	(155)
8.4.3 堆排序 .....	(156)
8.5 归并排序 .....	(159)
8.6 各种排序方法的综合比较 .....	(161)
8.7 外部排序 .....	(162)
8.8 各类排序算法的综合实现 .....	(162)
上机实习 排序 .....	(174)
习题 8 .....	(174)
<b>第 9 章 查找</b> .....	(178)
9.1 基本概念 .....	(178)
9.2 静态查找表 .....	(179)
9.2.1 顺序查找法 .....	(180)
9.2.2 折半查找法 .....	(181)
9.2.3 分块查找法 .....	(184)
9.3 动态查找表 .....	(186)
9.4 哈希表 .....	(194)
9.4.1 基本概念 .....	(194)
9.4.2 哈希函数构造方法 .....	(195)
9.4.3 冲突处理方法 .....	(198)
9.4.4 哈希表查找 .....	(200)
9.5 查找表实现 .....	(203)
上机实习 查找表 .....	(213)
习题 9 .....	(213)
<b>参考文献</b> .....	(216)

# 第 1 章 绪 论

自 1946 年世界上第一台电子计算机 ENIAC<sup>①</sup>诞生开始，特别是近 20 年来计算机产业的飞速发展和应用已远远超出人们的预料。Moore（摩尔）定理<sup>②</sup>指出，微处理器的速度每过 18 个月就会翻一番，而半导体芯片的价格每隔 18 个月就会降低一半。Metcalfe（迈特考夫）<sup>③</sup>定理指出，网络的价值与上网人数的平方成正比。计算机技术已成为现代科学技术发展的重要支柱，并逐步渗透到人类生活的各个领域。计算机的应用已不再局限于科学计算，而更多地用于控制、管理及数据处理等非数值计算的处理工作。计算机软件是计算机系统的灵魂，没有软件的计算机系统无法工作。由于计算机软件的核心是算法，而算法实质上是加工数据的过程，所以研究数据结构对提高编程能力和设计高性能的算法至关重要。本章将介绍数据结构和算法分析的基本概念。

## 1.1 数据结构基本概念

### 1.1.1 数据结构实例

**【例 1.1】** 学生成绩登记表。

图 1.1 中的每一行可以当做一个结点（或称为记录），它由学号、姓名、性别、3 门课程的成绩、总分等数据项组成。在这个表中，第一个记录没有直接前驱，称为开始结点；最后一个记录没有直接后继，称之为终端结点。除了第一个记录和最后一个记录以外，其余记录都只有一个直接前驱记录和一个直接后继记录。这种结点之间的关系是一一对应的，这种关系称为线性关系。根据数据之间的这种逻辑关系，我们可以定义给出学号查询该同学的成绩和在某个同学的后面插入另一个同学的成绩记录等运算。

学号	姓名	性别	操作系统	计算机网络	高等数学	总分
01	丁一	男	90	73	94	257
02	马二	男	67	75	91	233
03	张三	女	87	91	67	245
04	李四	男	90	89	79	258
05	王五	女	97	85	85	267
06	赵六	男	78	93	91	262
...	...	...	...	...	...	...

图 1.1 学生成绩登记表

① 第二次世界大战期间，美国军方为了解决计算大量军用数据的难题，成立了由宾夕法尼亚大学的莫奇利和埃克特领导的研究小组，于 1946 年 2 月研制成功第一台计算机 ENIAC。

② 由 Intel 公司创始人摩尔 1976 年提出。

③ 以太网的发明人。

### 【例 1.2】井字棋对弈问题。

图 1.2 (a) 是井字棋对弈过程中的一个格局，任何一方只要使相同的 3 个子连成一条直线（可以是一行、一列、或一条对角线）即为胜方。如果下一步由“×”方下，可以派生出 5 个子格局，如图 1.2 (b)；随后由“○”方接着下，对于每个子格局又可以派生出 4 个子格局……。

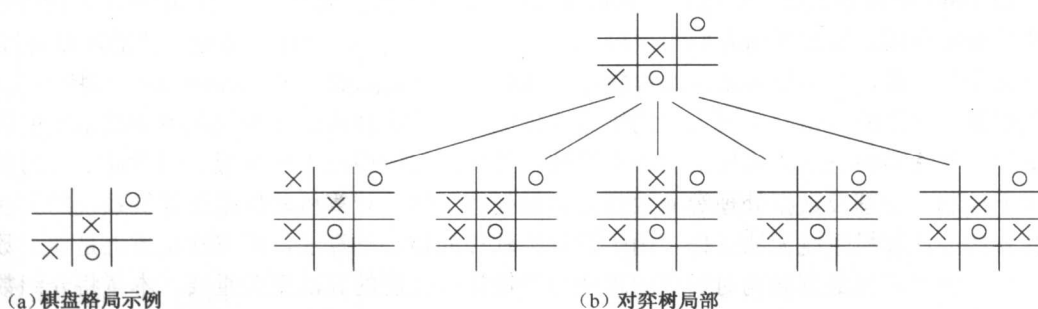


图 1.2 井字棋对弈“树”

若将从对弈开始到结束的过程中所有可能的格局画在一张图上，即形成一棵倒挂的对弈“树”。“树根”是对弈开始之前的棋盘格局，而所有“叶子”便是可能出现的结局，对弈过程就是从树根沿树叉到某个叶子的过程。在本例中，对弈中出现的每一个棋盘格局都可以算成一个“结点”。对弈开始之前的棋盘格局没有直接前驱，称为开始结点（根），以后每走一步棋，都有多种应对的棋盘格局，结点之间存在着一对多的关系，这种关系称为“树”关系。根据这种逻辑关系，我们可以定义求解某个棋局有多少个后续棋局，或者求解某个棋局是从哪一个棋局发展而来的等运算。

### 【例 1.3】7 桥问题。

流经哥尼斯堡的普雷格河的河湾处有两个小岛，由 7 座桥连接着两岸和小岛。当地流传一个游戏，要求在一次散步中恰好通过每座桥一次，再回到原出发点。假设 4 块陆地分别为 A, B, C, D，如图 1.3 所示。

1736 年，瑞士数学家 Euler（欧拉）引入图的概念，证明 7 桥问题永远无解。欧拉以结点表示 4 块陆地，以边表示桥，把图 1.3 抽象为图 1.4，即欧拉回路。这样，上述游戏变成欧拉回路能否一笔画成的问题。欧拉对 7 桥问题的结论是：“所有结点的度（一个结点拥有的边数称为度）均为偶数时，原问题才有解”。我们不想讨论这一数学证明问题，我们感兴趣的是在图 1.4 中，每一个结点都有多个直接前驱和多个直接后继。也就是说，结点与结点之间存在多对多关系的这样一种数据结构，这种关系被称为“图”关系。根据这种逻辑关系，我们可以定义与某一结点相邻的结点有哪些，两个结点间有多少条通路等运算。

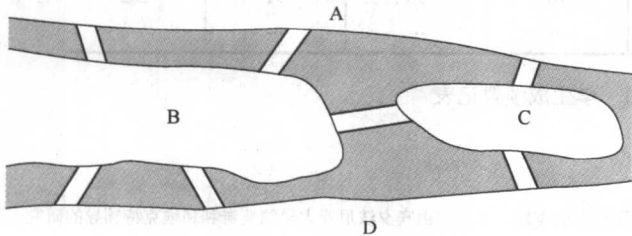


图 1.3 7 桥问题

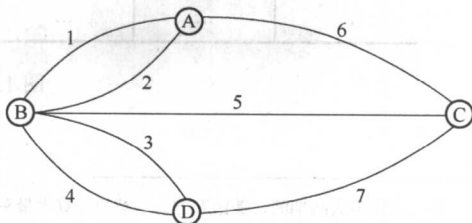


图 1.4 欧拉回路

综上所述,非数值计算问题的数学模型不再是数学方程,而是要研究诸如例 1.1、例 1.2 和例 1.3 所描述的数据之间的这些关系,以及通过数据之间的这些关系可以方便地进行何种运算。因此,简单地说数据结构是一门研究数据之间的关系及其运算的学科。

数据结构作为一门独立的学科始于 1968 年,但在此之前有关内容已散见于编译原理和操作系统的教材之中。1968 年,美国的图灵奖<sup>①</sup>获得者克努特(D. E. Knuth)教授<sup>②</sup>开创了数据结构的最初体系,他的著作《计算机程序设计艺术》第 1 卷《基本算法》是第一本比较系统地阐述数据的逻辑结构和存储结构及其操作的著作。20 世纪 60 年代末到 70 年代,出现了大型程序,软件也相对独立,结构程序设计即成为程序设计的主要内容。人们越来越重视数据结构,著名的瑞士计算机科学家沃斯(N. Wirth)教授<sup>③</sup>指出,算法+数据结构=程序。70 年代中期到 80 年代初,各种版本的数据结构著作相继出现。目前,数据结构的发展并未终结,一方面,面向各专门领域中特殊问题的数据结构得到研究和开发,例如多维图形数据结构等;另一方面,从抽象数据类型和面向对象的观点来讨论数据结构已成为一种新的趋势,越来越为人们所重视。

从我国计算机教学现状来看,数据结构不仅是计算机专业教学计划中的核心课程之一,而且已逐步成为非计算机专业的主要选修课程之一。数据结构又是一门介于数学、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程序设计的基础,而且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。打好数据结构课程的扎实基础,对于学习计算机专业的其他课程,例如操作系统、编译原理、数据库管理系统、软件工程和人工智能等都是十分有益的。

## 1.1.2 数据结构概念

### 1. 数据

数据是信息的载体,是对客观事物的符号表示。通俗地说,凡是能被计算机识别、存取和加工处理的符号、字符、图形、图像、声音、视频信号、程序等一切信息都可以称为数据。数据可以是数值数据,也可以是非数值数据。数值数据包括整数、实数、浮点数、复数等,主要用于科学计算和商务处理等;非数值数据包括文字、符号、图形、图像、动画、语音、视频信号等。随着多媒体技术的飞速发展,计算机中处理的非数值数据越来越多。

### 2. 数据元素

数据元素是对现实世界中某独立个体的数据描述,是数据的基本单位。在计算机中,数据元素通常作为一个整体来处理。一个数据元素可以由若干个数据项组成,在 C 程序设计中一个数据元素可以由一个 struct 表示。数据项是具有独立意义的最小数据单位,是对数据元素属性的描述。例如在图 1.1 中,每个数据元素由学号、姓名、性别、操作系统、计算机网

① 图灵奖由美国计算机学会(ACM)于 1966 年创立,以计算机概念的创始人图灵的名字命名,每年评选出 1~3 名获奖者,是世界计算机科学领域的最高奖项,有该领域的诺贝尔奖之称。

② 斯坦福大学教授,1974 年度图灵奖获得者,所著《计算机程序设计艺术》被誉为“计算机的圣经”。

③ Pascal 语言之父及结构化程序设计的首创者,1984 年度图灵奖获得者。

络、高等数学和总分等数据项组成。

### 3. 数据对象

数据对象是具有相同性质的数据元素的集合，是数据的一个子集。例如，字母字符数据对象是集合  $C = \{ 'A', 'B', 'C', \dots, 'Z' \}$ 。

### 4. 数据结构

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。根据数据元素之间的关系特性，数据结构可由一个二元组  $(D, S)$  定义，其中  $D$  是数据元素的有限集， $S$  是  $D$  种元素的关系的有限集。通常，基本的数据结构有如下 3 类。

- (1) 线性结构。结构中的数据元素之间存在一个对一个的关系。如例 1.1。
- (2) 树型结构。结构中数据元素之间存在一个对多个的关系。如例 1.2。
- (3) 图结构或网结构。结构中的数据元素之间存在多个对多个的关系。如例 1.3。

图 1.5 描述了这 3 种数据结构。3 种数据结构可以划分为两大类：线性结构和非线性结构。线性结构的逻辑特征是，有且仅有一个开始结点和一个终端结点，并且所有的结点都最多只有一个直接前驱和一个直接后继。非线性结构的逻辑特征是一个结点可能有多个直接前驱和直接后继。非线性结构包括树型结构和图结构。

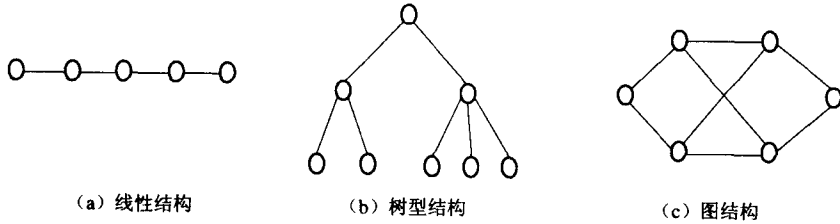


图 1.5 3 类基本数据结构示意图

### 5. 逻辑结构

逻辑结构描述的是数据元素之间的逻辑关系。通常说的线性结构、树型结构和图结构就是指数据元素的逻辑结构。

### 6. 物理结构

物理结构是数据结构在计算机中的表示，又称为存储结构。具有某种逻辑结构的数据元素在计算机中有顺序映像和非顺序映像两种不同的表示方法，由此得到两种不同的存储结构——顺序结构和链式结构。顺序结构的特点，是借助于数据元素在存储器中的相对位置表示数据元素之间的逻辑关系。链式结构的特点是，借助于指示数据元素存储地址的“指针”表示数据元素之间的逻辑关系。这两种存储结构可以用图 1.6 来描述。假如要存储字母表  $L = \{ 'A', 'B', 'C', \dots, 'Z' \}$ ，它的存储结构可以采取图中的两种形式。数据的逻辑结构和物理结构是密切相连的两个方面，依据数据的逻辑

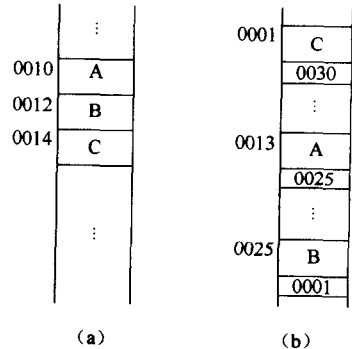


图 1.6 顺序存储和链式存储结构示意图

结构可以设计合理的算法，而算法的实现要依据数据的存储结构。

## 1.2 算法分析基本概念

### 1.2.1 算法

算法是解决特定问题的方法，是对数据施加的一系列操作。严格地说，算法是由若干指令组成的有限序列。程序是算法的实现。根据实际的需要，一个实际问题的解决可以有多种算法。算法包括数值算法和非数值算法两种。解决数值问题的算法称为数值算法，例如求解线性方程组，求解代数方程，求解微分方程等。解决非数值问题的算法称为非数值算法，数据处理方面的算法都属于非数值算法，例如各种排序算法、查找算法、插入算法、删除算法、遍历算法等。

一种算法必须具有以下5个特性。

- (1) 有穷性。一个算法包括的指令数必须有限，每一条指令的执行次数也必须有限。
- (2) 确定性。算法的每一条指令必须有确切的定义，无二义性。
- (3) 可行性。算法中的每一条指令都可以通过有限次、可实现的基本运算且在有限的时间内实现。
- (4) 输入。一个算法具有零个或多个输入。
- (5) 输出。一个算法具有一个或多个输出。

算法设计的要求如下。

- (1) 正确性。算法的执行结果应当满足预先设定的功能和要求。在实际应用中，算法的“正确性”有多个层次的含义。
- (2) 可读性。一个算法应当思路清晰、层次分明、易读易懂，有利于人对算法的理解。
- (3) 健壮性。当输入非法数据时，应能作适当反应和处理，不至引起莫名其妙的后果。
- (4) 高效性和低存储量。对同一个问题，执行时间越短，算法的效率就越高；完成相同的功能，执行算法时所占用的存储空间应尽可能少。

实际上，一种算法很难做到十全十美，原因是上述要求有时是相互抵触的。例如，时间和空间就是一对矛盾，如果要节约算法的执行时间，往往以牺牲一定存储空间为代价；而为了节省存储空间就可能耗费更多的计算时间。所以，实际操作中应以算法的正确性为前提，根据具体情况而有所侧重。若一个程序使用的次数较少，一般要求简明易懂即可；对于需要反复多次使用的程序，应尽可能选用快速的算法；若待解决的问题数据量极大，而计算机的存储空间又相对较小，则应主要考虑如何节约存储空间。

### 1.2.2 算法效率分析

算法执行时间需要根据该算法编制的程序在计算机上的执行时间来确定。一个算法所耗费的时间等于算法中每条语句的执行时间之和。每一条语句的执行时间是该语句的执行次数（frequency count，频度）与该语句执行一次所需时间的乘积，而每条语句的执行时间取决于其对应机器指令的执行时间。一个使用高级程序语言编写的程序在计算机上运行所需时间取决于如下因素。

- (1) 使用何种程序设计语言。实现编程语言的级别越高，其执行效率就越低。
- (2) 选用何种策略的算法。
- (3) 算法涉及问题的规模（求解问题的输入量，通常用  $n$  表示）。例如，求 50 以内的

素数与求 1000 以内的素数的执行时间必然是不同的。

(4) 编译程序所生成目标代码的质量。对于代码优化较好的编译程序, 所生成的程序质量较高。

(5) 机器执行指令的速度。

(6) 计算机的体系结构。并行计算通常能缩短算法的计算时间。

显然, 在各种因素不确定的情况下, 使用执行算法的绝对时间来衡量算法的效率是不合适的。在上述各种与计算机相关的软、硬件因素确定以后, 一个特定算法的运行时间就只依赖于问题的规模 (通常用正整数  $n$  表示)。

### 1.2.3 算法效率评价

算法的效率通常用时间复杂度和空间复杂度来评价, 它们反映了算法的执行所需要的时间和空间与问题规模之间的一种数量上的依赖关系。

#### 1. 时间复杂度

通常, 使用算法中所有语句的频度之和  $f(n)$  表示该算法所需的时间。在假定执行一条语句需要的时间固定的情况下, 语句的数量能大致表示算法运行所需的时间, 因此  $f(n)$  粗粒度地描述了一个算法所需时间。假设当问题规模  $n$  趋向于无穷大时, 有下式成立:

$$T(n) = O(f(n)) \quad (1.1)$$

它表示随着问题规模的扩大,  $T(n)$  的增长率和  $f(n)$  的增长率相同。称  $T(n)$  为算法的渐近时间复杂度 (asymptotic time complexity), 简称为时间复杂度 (time complexity)。

**【例 1.4】** 交换 A 和 B 的内容。

- (1) Temp=A;
- (2) A=B;
- (3) B=Temp;

这 3 条语句的频度都是 1,  $f(n)=3$ 。所以, 该程序的执行时间与问题规模无关,  $f(n)=3$ 。算法的时间复杂度为常数阶, 记为  $T(n)=O(1)$ 。

**【例 1.5】** 累加。

- (1) x=0; /\*执行 1 次\*/
- (2) y=0; /\*执行 1 次\*/  
for(k=1;k<=n; k++)
- (3) x++; /\*执行 n 次\*/  
for(i=1;i<=n; i++)  
for(j=1;j<=n; j++)
- (4) y++; /\*执行  $n^2$  次\*/

所有语句的频度之和  $f(n)$  为  $n^2+n+2$ 。

当  $n \rightarrow \infty$  时, 显然有

$$\lim f(n)/n^2 = \lim (n^2+n+2)/n^2 = 1$$

所以,

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

①  $O$  的数学定义是, 若  $f(n)$  为正整数  $n$  的函数, 则  $x = O(f(n))$  表示存在一个正常数  $M$ , 使得  $n \geq n_0$  时满足  $|x_n| \leq M|f(n)|$ 。

将常见的时间复杂度按数量级递增排列, 则依次为: 常数阶  $O(1)$ , 对数阶  $O(\lg n)$ , 线性阶  $O(n)$ , 线性对数阶  $O(n \lg n)$ , 平方阶  $O(n^2)$ , 立方阶  $O(n^3)$  和指数阶  $O(2^n)$ 。数量级越高, 说明算法所需的时间随问题规模的增大而以更高的速度增加。一个算法的时间复杂度最好是常数阶, 最高不超过  $k$  次幂阶。如果算法的时间复杂度为指数阶, 则该算法无法使用, 因为时间会随问题规模的增大而以指数增长。

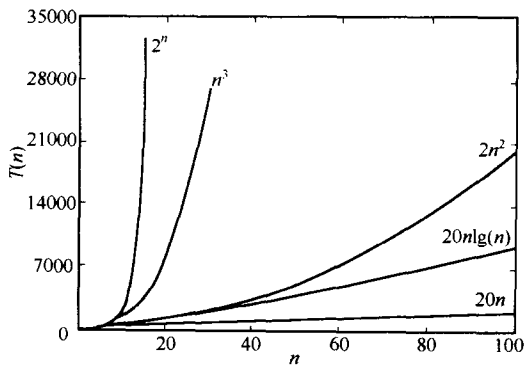


图 1.7 常见函数的增长率

## 2. 空间复杂度

一个程序的空间复杂度 (space complexity) 是指运行完一个程序所需的内存容量, 即存储量。利用程序的空间复杂性, 可以对该程序的运行所需内存有个预先的估计。程序运行所需的存储空间包括以下两部分。

### (1) 固定部分

这部分空间与所处理数据的大小和个数无关, 或者称与问题实例的特征无关。它主要包括指令空间 (即代码空间)、数据空间 (常量、简单变量、定长复合变量所占的空间)。

### (2) 可变部分

它由以下部分组成: 复合变量所需的空间 (这些变量的大小依赖于所使用的算法), 动态分配的空间, 以及递归栈所需的空间。这部分空间的大小与算法有关。

算法所需存储空间的用  $f(n)$  表示, 即

$$S(n) = O(f(n)) \quad (1.2)$$

其中,  $n$  为问题的规模,  $S(n)$  为算法的空间复杂度。

在进行时间复杂度和空间复杂度分析时, 如果所需时间和所占空间量都依赖于特定的输入, 一般都按最坏情况分析。

## 习题 1

### 一、填空题

1. 数据元素的逻辑结构包括: \_\_\_\_\_、\_\_\_\_\_ 和 \_\_\_\_\_ 3 种类型, 树型结构和图结构合称为 \_\_\_\_\_。

2. 线性结构中的元素之间存在 \_\_\_\_\_ 的关系, 树型结构中的元素之间存在 \_\_\_\_\_ 的关系, 图结构的元素之间存在 \_\_\_\_\_ 的关系。



3. 在树型结构中, 根结点\_\_\_\_\_前驱结点, 其余每个结点有且仅有\_\_\_\_\_个前驱结点; 叶子结点\_\_\_\_\_后继结点, 其余每个结点都可以有\_\_\_\_\_个后继结点。
4. 在图结构中, 每个结点的前驱结点可以有\_\_\_\_\_, 后继结点可以有\_\_\_\_\_。
5. \_\_\_\_\_是数据的基本单位, \_\_\_\_\_是数据不可分割的最小单位。
6. 算法的 5 个特性是\_\_\_\_\_, \_\_\_\_\_、\_\_\_\_\_, \_\_\_\_\_和\_\_\_\_\_。
7. 数据结构是一门研究非数值计算的程序设计问题中计算机的\_\_\_\_\_, 以及它们之间的\_\_\_\_\_和运算等的学科。
8. 任何一个算法的设计取决于选定数据的\_\_\_\_\_, 而算法的实现依赖于所采用的\_\_\_\_\_。
9. 数据元素之间的关系在计算机中有两种不同的表示方法, 即\_\_\_\_\_和\_\_\_\_\_。

## 二、选择题

1. 数据结构通常研究数据的\_\_\_\_\_及运算。
  - A. 物理结构和逻辑结构
  - B. 存储和抽象
  - C. 理想和抽象
  - D. 理想与逻辑
2. 数据结构中, 在逻辑上可以把数据结构分成\_\_\_\_\_。
  - A. 动态结构和静态结构
  - B. 紧凑结构和非紧凑结构
  - C. 线性结构和非线性结构
  - D. 内部结构和外部结构
3. 数据在计算机存储器内表示时, 如果元素在存储器中的相对位置能反映数据元素之间的逻辑关系, 则称这种存储结构为\_\_\_\_\_。
  - A. 存储结构
  - B. 逻辑结构
  - C. 顺序存储结构
  - D. 链式存储结构
4. 线性结构是指数据元素之间存在一种\_\_\_\_\_。
  - A. 一对多关系
  - B. 多对多关系
  - C. 多对一关系
  - D. 一对一关系
5. 在非线性结构中, 每个结点\_\_\_\_\_。
  - A. 无直接前驱
  - B. 只有一个直接前驱和个数不受限制的直接后继
  - C. 只有一个直接前驱和后继
  - D. 有个数不受限制的直接前驱和后继
6. 若  $f(n)=3n^2+2n+1$ , 则  $f(n)=$ \_\_\_\_\_。
  - A.  $O(n^2)$
  - B.  $O(n)$
  - C.  $O(2n)$
  - D.  $O(3n^2)$

## 三、分析题

试分析下列程序段的语句频度和时间复杂度。

1. 

```
for(i=0; i<n; i++)
  for(j=0; j<m; j++)
    A[i][j];
```
2. 

```
i=s=0;
while(s<n)
{
  i++;
  s+=i;
}
```
3. 

```
s=0;
for(i=0; i<n; i++)
```