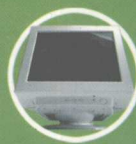


高等院校计算机系列教材

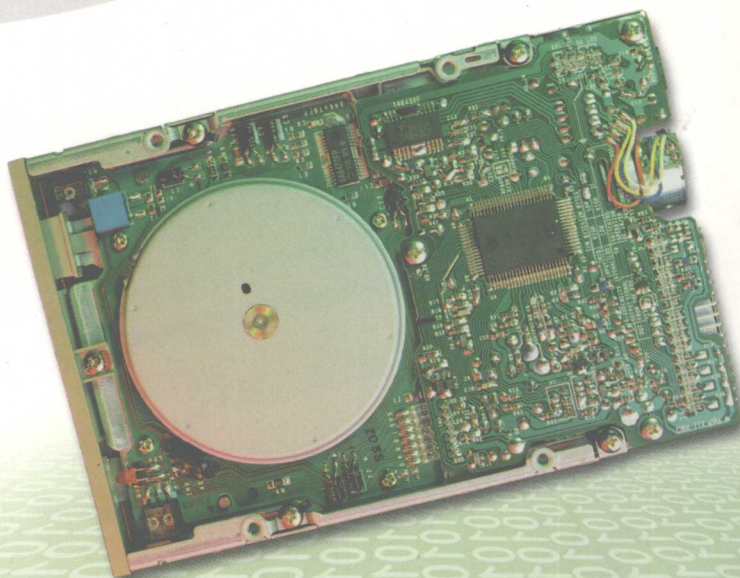


GAODENGYUANXIAO  
JISUANJI  
XILIEJIAOCAI



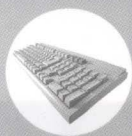
# 数据结构与算法

总主编：陈火旺 主编：谭骏珊 杨卫民 刘军万 湖南省计算机学会规划教材 中南大学出版社

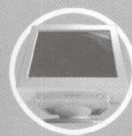


SJJGYSF  
SHUJUJIEGOU  
YUSUANFA

高等院校计算机系列教材



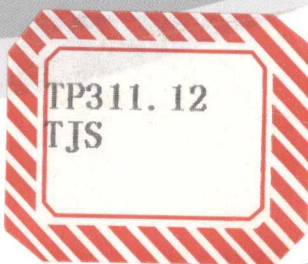
GAODENGYUANXIAO  
JISUANJI  
XILIEJIAOCAI



# 数据结构与算法

总主编：陈火旺 湖南省计算机学会规划教材 中南大学出版社

TP311.12  
TJS



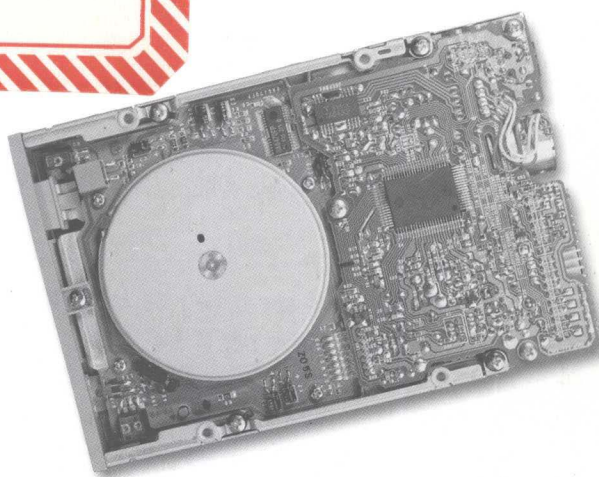
主 编：谭骏珊 杨卫民 刘军万

副主编：莫 照

编 委：（按姓氏笔画排序）

陈超云 陈灵娜 李盛欣 周咏梅

徐洪智



---

图书在版编目(CIP)数据

数据结构与算法/谭骏珊主编. —长沙:中南大学出版社,  
2005. 8

ISBN 7-81105-158-3

I. 数... II. 谭... III. ①数据结构②算法分析 IV. TP311.12

中国版本图书馆 CIP 数据核字(2005)第 094139 号

---

数据结构与算法

主编 谭骏珊 杨卫民 刘军万

---

☐责任编辑 马 泓 谭晓萍

☐责任印制 文桂武

☐出版发行 中南大学出版社

社址:长沙市麓山南路

邮编:410083

发行科电话:0731-8876770

传真:0731-8710482

☐印 装 中南大学印刷厂

---

☐开 本 787 × 1092 1/16 ☐印张 15.25 ☐字数 364 千字 插页 2

☐版 次 2005 年 8 月第 1 版 ☐2005 年 8 月第 1 次印刷

☐书 号 ISBN 7-81105-158-3/TP · 021

☐定 价 22.00 元

---

图书出现印装问题,请与出版社调换

# 高等院校计算机系列教材编委会

总 主 编 陈火旺

执行总主编 孙星明

副 总 主 编 李仁发 陈志刚

编 委 (按姓氏笔画排序)

|     |     |     |     |
|-----|-----|-----|-----|
| 王志英 | 刘任任 | 刘 宏 | 刘振宇 |
| 孙星明 | 羊四清 | 阳小华 | 阳爱民 |
| 余绍黔 | 吴宏斌 | 张新林 | 李仁发 |
| 李正华 | 李 军 | 李勇帆 | 李 峰 |
| 杨路明 | 沈 岳 | 肖建华 | 肖晓丽 |
| 陈火旺 | 陈志刚 | 罗庆云 | 金可音 |
| 胡志刚 | 赵 欢 | 徐建波 | 殷建平 |
| 郭国强 | 高守平 | 庾 清 | 黄国盛 |
| 龚德良 | 傅 明 | 彭民德 | 曾碧卿 |
| 蒋伟进 | 鲁荣波 | 谭骏珊 | 谭敏生 |

# 总 序

21 世纪,人类社会已经步入信息时代,信息产业推动着全球经济的蓬勃发展,改变着人类的联系与交换方式,从某种意义上说,信息革命是人类历史上又一次深刻的社会变革。无疑,在以信息产业为基础的知识经济社会中,计算机科学与技术具有举足轻重的地位。有鉴于此,当今世界各国皆把培养高素质的创新型计算机科学与技术专业人才作为一项重要的战略任务来抓。早在 1984 年,邓小平同志就强调指出:“计算机的普及要从娃娃抓起。”从此开启了中国信息革命的征程。经过 20 多年的努力,我国的计算机教育虽然取得了令人瞩目的成就,但离知识经济社会的要求还有很大的差距。据 2005 年信息产业部的数据显示,我国的信息化人才资源指数仅为 13.43,每年短缺信息化专业人才达 100 万之多。因此,快速培养和造就一大批高素质的计算机与信息人才,乃是我国高等教育所面临的一项严峻挑战。为此,我们必须改革和完善现有计算机与信息技术学科的教学计划和课程体系,优化课程结构,精炼教学内容,拓宽专业基础,强化实践环节,注重学生的知识、能力和综合素质的培养。

为了适应计算机科学与技术学科发展和教育的需要,湖南省计算机学会,参照《中国计算机科学与技术学科教程 2002》,组织了一批长期从事计算机科学与技术专业教学与科研的学者参与编撰了这套由中南大学出版社出版的《高等院校计算机系列教材》,希望在教材中及时反映学科前沿的研究成果与发展趋势,以高水平的科研促进教材建设,以优秀教材促进教学质量的提高。该系列教材具有如下特点:

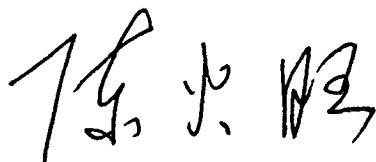
1. 教材参照《中国计算机科学与技术学科教程 2002》建议的教学大纲、知识领域、知识单元和知识点,结合作者多年教学与科研经验来编写,注重基本理论、基础知识的梳理、推演与挖掘,注意知识的更新,跟踪新技术、新成果的发展,并将之吸收到教材中来,力求开阔学生视野,逐步形成“基础课程精深,专业课程宽新”的格局,努力提高教材质量。

2. 注重理论联系实际,注意能力的培养。力图通过案例教学、课堂讨论、课程实验设计与实习,训练学生掌握知识、运用知识分析并解决实际问题的能力以满足学生今后从事科研和就业的需要。

3. 在规范教材编写体例的同时,注重写作风格的灵活性:每册的每个章节包括教学目的、本章小结、思考题与练习题,每门教材都配有 PPT 电子教案,并做到层次分明、逻辑性强、概念清楚、图文并茂、表达准确、可读性强。

这套教材的编写吸纳了广大计算机科学与技术教育工作者多年的教学与科研成果，凝聚了作者们的辛勤劳动，也得到了湖南省各高等院校相关专业领导和专家的大力支持。我相信这套教材的出版，对我国计算机科学与技术专业本科教学质量的提高将有很好的促进作用。

由于编委和作者们水平与时间的限制，教材中难免还有不足之处，恳请广大读者批评指正。

A handwritten signature in black ink, reading '陈海明' (Chen Haiming). The characters are written in a fluid, cursive style.

2005 年 7 月

## 前言

数据结构课程是计算机科学与技术专业的一门核心课程,是操作系统、数据库、软件工程、编译原理等课程的基础。现也成为了非计算机类学生学习计算机的必修课。计算机在各个领域的应用过程中,都会用到各种各样的数据结构,特别是各种特殊数据表示,就更需要学会研究计算机数据处理对象的特性、数据之间的逻辑关系、存储方式以及相应操作的实现。通过数据结构课程的学习,能够给读者提供软件开发和程序设计必要的技能训练,可进一步提高软件设计和编程水平。同时,通过对不同存储结构和相应算法的对比与编程练习,可以增强读者根据求解问题的性质选择合理的数据结构并控制求解算法的空间、时间复杂性的能力。

本书介绍了各种最常用的数据结构,针对每种数据结构从它的逻辑结构、存储方式以及这些数据结构的基本操作的实现三个方面进行了阐述,并对算法的效率进行了简要的分析和讨论。

本书共分10章。第1章介绍了数据结构课程所涉及的基本概念和基本知识,并给出了算法的描述方法和算法分析的方法;第2章介绍了线性表的定义,重点讨论了线性表在顺序存储结构和链式存储结构上各种基本运算的实现;第3章介绍了栈和队列两种特殊线性表的逻辑结构和运算,重点介绍了栈和队列在两种存储结构上的实现,并通过实例介绍了它们的应用;第4章介绍了有关串的基本概念、存储方式以及运算的实现,重点讨论了模式匹配算法;第5章介绍了数组和广义表的存储结构以及相关运算的实现;第6章介绍了树的基本概念、存储结构、树和二叉树的转换,重点讨论了二叉树的性质、存储结构、遍历方法和线索化二叉树;第7章介绍了图的有关概念、图的存储结构、遍历,重点讨论了图的应用:最小生成树、拓扑排序、关键路径和最短路径;第8章介绍了各种查找方法的基本思想以及各种查找方法的实现;第9章介绍了各种排序方法的基本思想和实现;第10章对有关文件的基本知识作了简要介绍。

本书内容丰富,条理清晰,叙述易懂,算法简单,可读性强。每章附有本章小结与习题,帮助读者掌握学习要点以及理解、巩固学习内容。本书既可作为高等学校计算机科学与技术专业的数据结构课程的教材,也可供广大从事计算机科学工作的科技人员自学参考。

本书由湖南省计算机学会组织,由中南林学院谭骏珊、杨卫民组织人员编写,具体分工

如下：陈超云（第1章）；陈灵娜（第2章）；周咏梅（第3章）；李盛欣（第4章）；刘军万（第5章）；杨卫民（第6章、第10章）；谭骏珊（第7章）；莫照（第8章）；徐洪智（第9章）。最后由主编统稿、修改、定稿。书中参考、借鉴了国内外同类教材和专著，在此一并表示感谢。

计算机技术发展日新月异，许多新概念、新技术都处在不断的发展之中，由于作者学识和水平有限，加之时间仓促，书中肯定存在错误或不当之处，敬请广大读者和专家批评指正。

编 者

2005年6月

## 目 录

|                          |      |
|--------------------------|------|
| 第1章 绪 论 .....            | (1)  |
| 1.1 为什么要学习数据结构 .....     | (1)  |
| 1.2 基本概念和术语 .....        | (4)  |
| 1.3 抽象数据类型及面向对象的概念 ..... | (5)  |
| 1.4 算法和算法评价 .....        | (10) |
| 1.4.1 算法 .....           | (10) |
| 1.4.2 算法设计的要求 .....      | (11) |
| 1.4.3 算法效率的度量 .....      | (11) |
| 本章小结 .....               | (15) |
| 习 题 .....                | (15) |
| 第2章 线性表 .....            | (17) |
| 2.1 线性表的定义和抽象数据类型 .....  | (17) |
| 2.2 线性表的顺序表示及操作实现 .....  | (20) |
| 2.3 线性表的链式表示和实现 .....    | (25) |
| 2.3.1 线性链表 .....         | (26) |
| 2.3.2 循环链表 .....         | (34) |
| 2.3.3 双向链表 .....         | (35) |
| 2.4 线性表的应用 .....         | (37) |
| 本章小结 .....               | (40) |
| 习 题 .....                | (41) |
| 第3章 栈和队列 .....           | (43) |
| 3.1 栈 .....              | (43) |
| 3.1.1 抽象数据类型栈的定义 .....   | (43) |
| 3.1.2 栈的表示和实现 .....      | (45) |
| 3.1.3 栈的链式存储 .....       | (47) |
| 3.2 栈的应用举例 .....         | (48) |
| 3.2.1 数制转换 .....         | (48) |
| 3.2.2 数学表达式计算 .....      | (49) |
| 3.2.3 迷宫求解 .....         | (50) |
| 3.3 栈与递归 .....           | (52) |
| 3.4 队列 .....             | (56) |
| 3.4.1 队列的定义 .....        | (56) |

---

|                     |               |       |
|---------------------|---------------|-------|
| 3.4.2               | 队列的实现 .....   | (57)  |
| 3.4.3               | 循环队列 .....    | (61)  |
| 3.4.4               | 队列的应用举例 ..... | (63)  |
| 本章小结                | .....         | (64)  |
| 习 题                 | .....         | (65)  |
| <b>第4章 串</b>        | .....         | (66)  |
| 4.1 串类型的定义          | .....         | (66)  |
| 4.1.1 串的基本概念        | .....         | (66)  |
| 4.1.2 串的抽象数据类型      | .....         | (67)  |
| 4.2 串的实现和表示         | .....         | (69)  |
| 4.2.1 定长顺序存储表示      | .....         | (69)  |
| 4.2.2 堆分配存储表示       | .....         | (71)  |
| 4.2.3 串的链式存储表示      | .....         | (72)  |
| 4.3 串的模式匹配算法        | .....         | (73)  |
| 4.3.1 串模式匹配的古典算法    | .....         | (74)  |
| 4.3.2 串模式匹配的 KMP 算法 | .....         | (75)  |
| 本章小结                | .....         | (81)  |
| 习 题                 | .....         | (81)  |
| <b>第5章 数组与广义表</b>   | .....         | (83)  |
| 5.1 数组的定义           | .....         | (83)  |
| 5.2 数组的顺序表示及操作的实现   | .....         | (85)  |
| 5.2.1 数组的顺序表示       | .....         | (85)  |
| 5.2.2 数组基本操作的实现     | .....         | (86)  |
| 5.3 矩阵的压缩存储         | .....         | (88)  |
| 5.3.1 特殊矩阵          | .....         | (88)  |
| 5.3.2 压缩存储          | .....         | (89)  |
| 5.4 稀疏矩阵            | .....         | (91)  |
| 5.4.1 稀疏矩阵的压缩存储     | .....         | (91)  |
| 5.4.2 稀疏矩阵的运算       | .....         | (95)  |
| 5.5 广义表             | .....         | (100) |
| 5.5.1 基本概念          | .....         | (100) |
| 5.5.2 存储结构          | .....         | (102) |
| 5.5.3 基本运算          | .....         | (103) |
| 本章小结                | .....         | (104) |
| 习 题                 | .....         | (104) |
| <b>第6章 树</b>        | .....         | (106) |
| 6.1 树的概念            | .....         | (106) |

|                          |              |
|--------------------------|--------------|
| 6.1.1 树的定义 .....         | (106)        |
| 6.1.2 树的抽象数据类型 .....     | (106)        |
| 6.1.3 树的表示 .....         | (107)        |
| 6.1.4 树的基本术语 .....       | (108)        |
| 6.2 二叉树 .....            | (109)        |
| 6.2.1 二叉树的定义 .....       | (109)        |
| 6.2.2 二叉树的性质 .....       | (110)        |
| 6.2.3 二叉树的存储结构 .....     | (113)        |
| 6.3 遍历二叉树 .....          | (114)        |
| 6.3.1 遍历二叉树的递归算法 .....   | (114)        |
| 6.3.2 遍历二叉树的非递归算法 .....  | (116)        |
| 6.4 线索二叉树 .....          | (118)        |
| 6.4.1 线索 .....           | (118)        |
| 6.4.2 中序线索化二叉树 .....     | (118)        |
| 6.5 树与森林 .....           | (121)        |
| 6.5.1 树的存储表示 .....       | (121)        |
| 6.5.2 森林、树与二叉树的转换 .....  | (123)        |
| 6.5.3 树和森林的遍历 .....      | (124)        |
| 6.6 哈夫曼树及应用 .....        | (125)        |
| 6.6.1 基本术语 .....         | (125)        |
| 6.6.2 哈夫曼树的构造算法 .....    | (126)        |
| 6.6.3 哈夫曼编码 .....        | (127)        |
| 本章小结 .....               | (128)        |
| 习 题 .....                | (128)        |
| <b>第7章 图</b> .....       | <b>(130)</b> |
| 7.1 图的概念 .....           | (130)        |
| 7.1.1 图的定义 .....         | (130)        |
| 7.1.2 图的基本术语 .....       | (131)        |
| 7.2 图的存储结构 .....         | (133)        |
| 7.2.1 邻接矩阵表示法 .....      | (133)        |
| 7.2.2 邻接表 .....          | (134)        |
| 7.2.3 十字链表 .....         | (136)        |
| 7.2.4 邻接多重表 .....        | (137)        |
| 7.3 图的遍历 .....           | (138)        |
| 7.3.1 深度优先搜索 .....       | (139)        |
| 7.3.2 广度优先搜索 .....       | (140)        |
| 7.4 图的连通性问题 .....        | (141)        |
| 7.4.1 无向图的连通分量和生成树 ..... | (141)        |
| 7.4.2 最小生成树 .....        | (141)        |

---

|            |                        |              |
|------------|------------------------|--------------|
| 7.5        | 活动网络 .....             | (146)        |
| 7.5.1      | 用顶点表示活动的网络 .....       | (146)        |
| 7.5.2      | 用边表示活动的网络 .....        | (148)        |
| 7.6        | 最短路径 .....             | (152)        |
| 7.6.1      | 从某个源点到其余各顶点的最短路径 ..... | (152)        |
| 7.6.2      | 每一对顶点之间的最短路径 .....     | (154)        |
|            | 本章小结 .....             | (156)        |
|            | 习 题 .....              | (157)        |
| <b>第8章</b> | <b>查 找 .....</b>       | <b>(159)</b> |
| 8.1        | 基本概念 .....             | (159)        |
| 8.2        | 静态查找表 .....            | (161)        |
| 8.2.1      | 静态查找表结构 .....          | (161)        |
| 8.2.2      | 顺序查找 .....             | (161)        |
| 8.2.3      | 二分查找 .....             | (162)        |
| 8.2.4      | 分块查找 .....             | (164)        |
| 8.3        | 树表的查找 .....            | (165)        |
| 8.3.1      | 二叉排序树 .....            | (165)        |
| 8.3.2      | 平衡二叉树(AVL 树) .....     | (170)        |
| 8.3.3      | B-树和 B+树 .....         | (173)        |
| 8.4        | 哈希表 .....              | (179)        |
| 8.4.1      | 哈希表与哈希法 .....          | (179)        |
| 8.4.2      | 常用哈希函数的构造方法 .....      | (180)        |
| 8.4.3      | 处理冲突的方法 .....          | (182)        |
| 8.4.4      | 哈希表的查找分析 .....         | (184)        |
|            | 本章小结 .....             | (186)        |
|            | 习 题 .....              | (186)        |
| <b>第9章</b> | <b>排 序 .....</b>       | <b>(188)</b> |
| 9.1        | 概述 .....               | (188)        |
| 9.2        | 插入排序 .....             | (189)        |
| 9.2.1      | 直接插入排序 .....           | (189)        |
| 9.2.2      | 二分插入排序 .....           | (190)        |
| 9.2.3      | 表插入排序 .....            | (191)        |
| 9.2.4      | 希尔排序 .....             | (193)        |
| 9.3        | 交换排序 .....             | (194)        |
| 9.3.1      | 冒泡排序 .....             | (194)        |
| 9.3.2      | 快速排序 .....             | (196)        |
| 9.4        | 选择排序 .....             | (197)        |
| 9.4.1      | 直接选择排序 .....           | (198)        |

|                            |              |
|----------------------------|--------------|
| 9.4.2 树型选择排序 .....         | (198)        |
| 9.4.3 堆排序 .....            | (200)        |
| 9.5 归并排序 .....             | (202)        |
| 9.6 基数排序 .....             | (204)        |
| 9.6.1 多关键字排序 .....         | (204)        |
| 9.6.2 链式基数排序 .....         | (205)        |
| 9.7 各种内排序方法的比较和选择 .....    | (208)        |
| 9.7.1 各种内排序方法的比较 .....     | (208)        |
| 9.7.2 各种内排序方法的选择 .....     | (208)        |
| 9.8 外部排序 .....             | (209)        |
| 9.8.1 外部排序的基本过程 .....      | (209)        |
| 9.8.2 多路平衡归并排序 .....       | (211)        |
| 9.8.3 置换 - 选择排序 .....      | (213)        |
| 9.8.4 最佳归并排序 .....         | (216)        |
| 本章小结 .....                 | (217)        |
| 习 题 .....                  | (217)        |
| <b>第 10 章 文 件</b> .....    | <b>(219)</b> |
| 10.1 文件及其逻辑特性 .....        | (219)        |
| 10.1.1 有关文件的基本概念 .....     | (219)        |
| 10.1.2 文件操作的类型 .....       | (220)        |
| 10.1.3 文件的存储结构 .....       | (220)        |
| 10.2 顺序文件 .....            | (221)        |
| 10.2.1 顺序文件的特点 .....       | (221)        |
| 10.2.2 顺序文件操作 .....        | (221)        |
| 10.3 索引文件 .....            | (222)        |
| 10.3.1 概 述 .....           | (222)        |
| 10.3.2 静态索引——ISAM 文件 ..... | (223)        |
| 10.3.3 动态索引——VSAM 文件 ..... | (225)        |
| 10.3.4 两种索引结构的比较 .....     | (227)        |
| 10.4 散列文件 .....            | (227)        |
| 10.5 多重链接表文件 .....         | (229)        |
| 10.6 倒排文件 .....            | (230)        |
| 本章小结 .....                 | (230)        |
| 习 题 .....                  | (231)        |
| <b>参考文献</b> .....          | <b>(232)</b> |

## 第1章 绪论

本章主要介绍数据结构研究的主要内容。数据、数据元素、数据对象、数据结构、存储结构和数据类型等是数据结构中涉及的基本概念。本章介绍了数据结构的形式描述,抽象数据类型的定义、表示和实现方法;算法的概念、描述方法、评价标准以及对算法进行时间复杂度和空间复杂度分析的方法。

### 1.1 为什么要学习数据结构

如今,计算机已深入到人类社会的各个领域。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。与此相应,计算机加工处理的对象由纯粹的数值发展到字符、表格和图像等各种具有一定结构的数据,所处理的数据规模也不断加大,并具有一定的关系,这就给程序设计带来一些新的问题。为了编写出一个“好”的程序,必须分析待处理对象的特性以及待处理对象之间存在的关系。这就是“数据结构”这门学科形成和发展的背景。

用计算机解决一个问题要经过以下三个步骤:

- (1) 分析实际问题,从其中抽象出一个适当的数学模型。
- (2) 设计或选择一个解此数学模型的算法。
- (3) 编程、调试、测试、修改,直至得到最终的解答。

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含有的关系,然后用数学的语言加以描述。例如,“鸡兔同笼”问题的数学模型是二元一次方程组,结构静力分析计算的数学模型是线性代数方程组。然而,如今更多的计算机应用是非数值计算问题,对其操作也不再是单纯的数值计算,而更多的是需要对其进行组织、管理和检索。这类问题无法用数学方程加以描述。下面请看三个例子。

**例 1-1** 学籍档案管理问题。假设一个学籍档案管理系统应包含表 1-1 所示的学生信息。如果有新生入学,需要将该生的信息插入表中,有学生毕业,则需删除学生的信息,如某学生转专业或留级,则需修改该学生的信息,除此之外,更多的操作是按条件检索某个学生的信息,等等。这类问题的数据特点是每个学生的信息占据一行,所有学生的信息按学号顺序依次排列构成一张表格,表中每个学生的信息依据学号的大小存在着一种前后关系,这就是我们所说的线性结构。这类问题的数学模型中,计算机处理的对象之间通常存在着一种最简单的线性关系,该表构成的文件及其之间的“线性”关系就构成该类问题的数学模型,这类数学模型可称为线性的数据结构。

表 1-1 学生基本情况表

| 学 号      | 姓 名 | 性 别 | 专 业      | 年 级    |
|----------|-----|-----|----------|--------|
| 20010201 | 李 军 | 男   | 计算机科学与技术 | 2001 级 |
| 20010301 | 王颜霞 | 女   | 电子信息工程   | 2001 级 |
| 20010302 | 张 三 | 男   | 电子信息工程   | 2001 级 |
| 20010703 | 孙 涛 | 男   | 电算会计     | 2001 级 |
| 20020218 | 单晓宏 | 男   | 计算机科学与技术 | 2002 级 |

例 1-2 输出  $n$  个对象的全排列。

输出  $n$  个对象的全排列可以使用图 1-1 所示的形式描述。

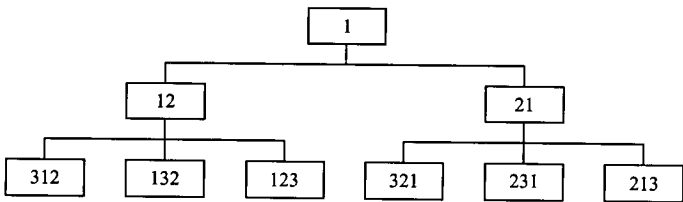


图 1-1 三个对象的全排列过程

在求解过程中，所处理的数据之间具有层次关系，这是我们所说的树型结构。对它的操作有：建立树型结构，输出最底层结点内容等。

例 1-3 课程考试时间安排问题。假设计算机系某学期为研究生开设了 6 门专业选修课：算法分析、形式语言、计算机图形学、模式识别、计算机网络、人工智能。5 名研究生的选课情况如表 1-2 所示。

表 1-2 研究生选课情况表

| 姓 名 | 选修课程 1 | 选修课程 2 | 选修课程 3 |
|-----|--------|--------|--------|
| 陈一  | 算法分析   | 形式语言   | 计算机网络  |
| 杨二  | 计算机图形学 | 模式识别   |        |
| 张三  | 计算机图形学 | 计算机网络  | 人工智能   |
| 李四  | 模式识别   | 人工智能   | 算法分析   |
| 王五  | 形式语言   | 人工智能   |        |

现要求设计一个考试日程安排表，使得在尽可能短的时间内安排完考试。要解决好这一问题，首先应选择一个合适的数据结构来表示它。为此，我们可以设计一个图，图中的顶点表示课程，在所有能同时考试的课程之间连上一条边。显然同一个学生选修的几门课是不能安排在同一时间考试的，因此该生所选修的课程中应该两两有边相连。由此可得如图 1-2

所示的数据结构模型。图中 A, B, C, D, E, F 分别对应 6 门课程。例如陈一选修的三门课是 A, B, E, 则 A 和 B, A 和 E, B 和 E 之间都有边相连。

上述的由顶点和边组成的无向图是数据结构中一类非线性的数据结构, 课程考试时间安排问题可以抽象为对无向图进行“着色”操作, 即用尽可能少的颜色去给图中的每个顶点着色, 使得任意两个相邻的顶点着上不同的颜色。每一种颜色表示一个考试时间, 着上同一颜色的顶点是可以安排在同一时间内考试的课程, 例如, 顶点 A 和 C 不相邻, 可选颜色 1 为它们着色。同理, B 和 D 可着同一颜色; E 和 F 相邻, 应分别着上不同颜色 3 和 4。也就是说, 只要安排 4 个时间考试即可, 时间 1 内可以考算法分析(A)和计算机图形学(C), 时间 2 内可以考形式语言(B)和模式识别(D), 时间 3 和时间 4 分别考计算机网络(E)和人工智能(F)。

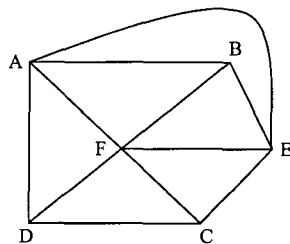


图 1-2 数据的图形结构示意图

由上述三个例子可知, 当今计算机的操作对象的关系更加复杂, 操作形式不再是单纯的数值计算, 而更多的是对这些具有一定关系的数据进行组织管理, 我们称其为非数值性处理, 描述这类非数值计算问题的数学模型不再是数学方程, 而是诸如表、树和图之类的数据结构。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。

“数据结构”作为一门独立的课程在国外是从 1968 年开始设立的。1968 年, 美国唐·欧·克努特教授开创了数据结构的最初体系, 他所著的《计算机程序设计技巧》第一卷《基本算法》是第一本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。从 20 世纪 60 年代末到 70 年代初, 出现了大型程序, 软件也相对独立, 结构程序设计成为程序设计方法学的主要内容。人们越来越重视数据结构, 认为程序设计的实质是对确定的问题选择一种好的结构, 加上设计一种好的算法。从 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。

在我国, “数据结构”也是计算机专业的教学计划中的核心课程之一。“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到计算机硬件(特别是编码理论、存储装置和存取方法等)的研究范围, 而且和计算机软件的研究有着更密切的关系, 无论是编译程序还是操作系统, 都涉及到数据元素在存储器中的分配问题, 需要用到大量的数据结构知识, 如操作系统中的作业队列、就绪进程队列等, 编译原理中借用栈来进行表达式求值等。在研究信息检索时也必须考虑如何组织数据, 以便查找和存取数据元素更为方便。因此, 可以认为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程序设计(特别是非数值计算的程序设计)的基础, 而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的重要基础。

数据结构课程主要是研究非数值计算的程序设计问题中所出现的计算机操作对象以及它们之间的关系和操作等。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发展, 如多维图形数据结构等; 另一方面, 随着面向对象程序设计技术

被广泛使用,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

数据(Data)是对客观事物的符号表示。在计算机科学中其含义是指所有能够输入到计算机中并被计算机程序处理的符号的总称。它是计算机程序加工的“原料”。例如,一个利用数值分析方法解代数方程的程序,其处理对象是整数和实数;一个编译程序或文字处理程序的处理对象是字符串。因此,对计算机科学而言,数据的含义极为广泛,如图像、声音等都可以通过编码而归之于数据的范畴。

数据元素(Data Element)是数据集合中的一个实体,是计算机程序中加工处理的基本单位。数据元素按其组成可分为简单型数据元素和复杂型数据元素。简单型数据元素由一个数据项组成,所谓数据项就是数据中不可再分割的最小单位;复杂型数据元素由多个数据项组成,它通常携带着一个概念的多方面信息。

数据对象是性质相同的数据元素的集合,是数据的一个子集。例如:整数的数据对象是 $\{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$ ,英文字母字符的数据对象是 $\{'A', 'B', 'C', 'D', 'E', \dots\}$ 。

数据结构(Data Structure),简单地说,就是相互之间存在一种或多种特定关系的数据元素的集合。在任何问题中,数据元素都不是孤立存在的,它们之间总是存在着某种关系,这种数据元素相互之间的关系称为结构。常见的数据结构有: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;②线性结构:结构中的数据元素之间存在一个对一的关系;③树形结构:结构中的数据元素之间存在一个对多的关系;④图状结构或网状结构:结构中的数据元素之间存在多个对多的关系。图1-3是上述四类结构的关系图。

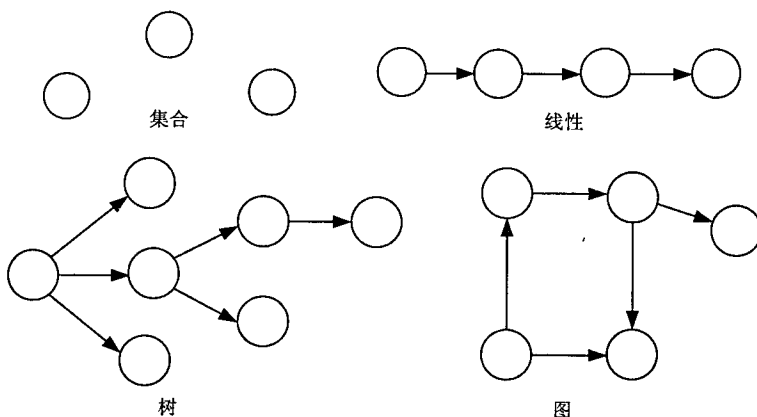


图 1-3 四类基本结构关系图

数据结构的定义形式为:数据结构是一个二元组

$$\text{Data\_Structures} = (D, S)$$