

高等院校计算机教育系列教材



数据结构教程

彭波 编著

01



清华大学出版社

高等院校计算机教育系列教材

数据结构教程

彭 波 编著

清华大学出版社
北 京

内 容 简 介

数据结构是计算机专业的重要基础课程,也是该专业的核心课程之一,它是一门集技术性、理论性和实践性于一体的课程。

本书介绍了抽象数据类型和基本数据结构,阐述了各种数据结构内在的逻辑关系,讨论了各种数据结构在计算机中的存储表示,给出了在各种数据结构上的基本运算及算法实现。内容包括:数据结构概述、线性表、栈和队列、串、多维数组与广义表、二叉树与树、图、查找表、内部排序、外部排序、文件和数据结构程序设计方法。书中使用类 C 语言作为算法描述语言,且所有算法都可以在任何一种 C 语言的开发环境中实现。书中每一章后面都配有适量的习题,以供读者复习提高自身水平。

本书可以作为高等院校计算机专业及相关专业的教材。对于计算机类专业的学生或从事计算机工程与应用工作的科技工作者,本书也是一本实用的参考教材。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

数据结构教程/彭波编著.—北京:清华大学出版社,2004

(高等院校计算机教育系列教材)

ISBN 7-302-08007-0

I.数… II.彭… III.数据结构—教材 IV.TP311.12

中国版本图书馆 CIP 数据核字(2004)第 004351 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦
http://www.tup.com.cn 邮 编:100084
社 总 机:010-62770175 客 户 服 务:010-62776969

组稿编辑:彭 欣

文稿编辑:刘 颖

封面设计:陈刘源

印 刷 者:北京国马印刷厂

装 订 者:三河市李旗庄少明装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:26.5 字数:627千字

版 次:2004年3月第1版 2004年3月第1次印刷

书 号:ISBN 7-302-08007-0/TP·5805

印 数:1~5000

定 价:34.00元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

丛 书 序

人类进入新千年时, 计算机科学已经具有一块极其活跃的、崇尚发展与创新的领地, 并成为我们这一时代决定性的先驱技术。计算机是现代文化构成不可或缺的组成部分, 是现代科学技术发展的先导, 并且是世界经济巨量增长的根本引擎。同时, 计算机技术的发展日新月异, 它的快速发展对计算机科学与工程的教育产生了深刻的影响。因此, 计算机教育在世界各国备受重视, 计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

伴随着计算机新技术的不断涌现, 已有技术几年内即变得陈旧。与此同时, 计算机教育也被其所在的专业、文化和社会范围的改变影响着。计算机学科已变得更宽广、内容更丰富, 其应用领域不断飞速增长。因此, 无论在教学体系、教学内容还是教学方法、教学手段上都必须进行深化改革, 与时俱进。CC2001 的出现, 无疑是对计算机学科课程体系一个崭新的完整的引导。我们工科院校无论计算机专业还是非计算机专业的计算机科学与工程的教育应该紧密有机地与工科学生的培养目标相结合, 注重知识、能力、素质教育三方面的综合教育, 加强计算机系统的认知、分析、设计和应用能力, 算法设计与分析能力, 程序设计能力以及计算思维能力等方面的培养。

原化工部部属高校计算机教育协作组结合工程教育的特点, 大力开展计算机教育协作与交流, 十年来开展了务实的、全方位的、卓有成效的教学研讨及教研观摩等活动, 极大地促进了交流并推动了各校计算机教育的发展。同时, 协作组不断地扩展, 吸收了许多其他领域的高校参加, 共同为我国计算机高等教育事业的发展与完善进行广泛的交流探索。

目前参加这个协作组的主要高校有:

| | | | |
|----------|--------|--------|----------|
| 清华大学 | 南京大学 | 天津工业大学 | 中国农业大学 |
| 北京化工大学 | 南京工业大学 | 青岛科技大学 | 郑州大学 |
| 武汉化工学院 | 沈阳化工学院 | 南京师范大学 | 华南理工大学 |
| 河北行政学院 | 南京工程学院 | 淮海工学院 | 北京石油化工学院 |
| 江苏石油化工学院 | | | |

在清华大学出版社的大力支持下, 本协作组 2001 年年会决定组织出版一套最新的计算机系列教材, 第一期出版 14 部有关程序设计与软件应用方面的教材。它们是: 《计算机导论》、《C 语言程序设计》、《Visual Basic 6.0 程序设计》、《Java 程序设计》、《面向对象程序设计——Visual C++》、《SQL Server 数据库原理及应用教程》、《C#编程及应用程序开发教程》、《组网技术与配置》、《数据库理论及应用基础》、《现代语音技术基础与应用》、《计算机图形学基础教程》、《Win32 汇编语言实用教程》、《C/C++程序设计教程》和《数据结构教程》等。

本系列教材依据 CC2001 框架，精心策划、准确定位，概念清晰，例题丰富，深入浅出，内容翔实，体系合理，重点突出，是一套面向高等学校计算机和非计算机专业学生的计算机基础与应用系列教材，也可供从事计算机应用和开发的各类人员学习使用。

本系列教材源于十几所全国重点大学和普通高等院校计算机教育的教学改革与实践，凝聚了工作在教学第一线的任课教师的教学经验与研究成果。我们期望本系列教材的出版，并在教学实践中不断完善与更新，为我国高校计算机教育事业做出新的贡献。

编委会

2003 年 12 月

编委会名单

主编：朱群雄

编委：张莉 闵华清 王晓峰

邵定宏 刘川来 彭四伟

刘斌 刘新民 张彦锋

吕纪国 刘焯 王相林

蔡莲红 孙正兴 冉林仓

彭波

前 言

“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,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。在研究信息检索时也必须考虑如何组织数据,以便使查找和存取数据元素更为方便。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个核心内容,是从事计算机科学研究及其应用的科技工作者必须掌握的重要内容。

本书在内容组织和编排上力求体现“先理论、后应用、理论与应用相结合”的原则,强调对理论知识的理解和运用。通过对本课程的学习,可掌握各种数据结构的基本概念、逻辑特性和物理表示法,以及相应运算的算法;灵活运用各种数据结构解决实际问题,并且为学习后继专业课程打下良好的基础。

全书的第1章综述数据、数据结构、抽象数据类型、算法描述及算法分析等概念;第2章~第7章从抽象数据类型的角度介绍了几种常用的数据结构,如线性表、栈、队列、串、多维数组、广义表、二叉树、树及图等,分别讨论了数据的逻辑结构和存储结构,以及相应运算的算法实现;第8章~第10章分别讨论了查找表、内部排序和外部排序的各种实现方法,并着重从时间上进行定性或定量的分析和比较;第11章介绍了常用的文件结构;第12章讲述了数据结构程序设计的方法,并以一个实例做进一步的说明。

全书采用了类C语言作为数据结构和操作算法的描述工具,它是C语言的一个精选子集,同时又采用了C++对C语言的非面向对象的增强功能。例如,动态分配和释放顺序存储结构的内存;利用引用参数传递函数运算的结果;使用默认参数以简化函数参数表的描述等。这些措施使数据类型的定义和数据结构相关操作算法的描述更加简明清晰、可读性更好,转变成C程序也极为方便。另一方面又奠定了一个基础,把类型定义和操作算法稍加技术处理,就很容易将其封装成类,并进一步转化成面向对象的程序模型。

本书可以作为高等院校计算机专业及相关专业的教材,教授时间可为80学时(包括讲课和上机)。为了便于读者理解,本书对数据结构众多知识点的来龙去脉做了详细的解释和说明,并配有大量的算法应用实例穿插其间。考虑到计算机技术的发展和进步,在内容编排方面尽量做到推陈出新,实例也力求新颖,以适应技术发展的潮流。

由于作者水平有限,书中难免有错误,请读者谅解,如果有问题需要与作者联系,请发送电子邮件到: pengbo_cau@sina.com。

在编写本书的过程中,孙一林、胡治国、吕小晴、崔永普、许振文、王茜、刘群、张伟娜等同志参加了算法的实现与调试,在此表示感谢。

编 者
2004.1

目 录

| | |
|--------------------------------|--|
| 第 1 章 绪论.....1 | |
| 1.1 数据结构的讨论范畴.....1 | |
| 1.2 数据结构的发展概况.....3 | |
| 1.3 数据结构的相关概念.....5 | |
| 1.3.1 基本概念和术语.....5 | |
| 1.3.2 数据结构.....6 | |
| 1.3.3 数据类型和抽象数据类型.....10 | |
| 1.4 数据结构的算法描述.....12 | |
| 1.4.1 算法.....12 | |
| 1.4.2 算法的描述.....13 | |
| 1.5 数据结构的算法分析.....17 | |
| 1.5.1 算法效率的度量.....17 | |
| 1.5.2 算法的空间需求.....19 | |
| 1.6 习题.....20 | |
| 第 2 章 线性表.....23 | |
| 2.1 线性表的类型定义.....23 | |
| 2.1.1 线性表的定义.....24 | |
| 2.1.2 线性表的抽象数据类型.....24 | |
| 2.2 线性表的顺序表示与实现.....27 | |
| 2.2.1 线性表的顺序存储表示.....28 | |
| 2.2.2 顺序表中基本操作的实现.....28 | |
| 2.2.3 顺序存储结构小结.....34 | |
| 2.2.4 应用举例.....35 | |
| 2.3 线性表的链式表示与实现.....38 | |
| 2.3.1 线性表的链式存储表示.....39 | |
| 2.3.2 单链表中基本操作的实现.....39 | |
| 2.3.3 循环链表.....44 | |
| 2.3.4 双向链表.....44 | |
| 2.3.5 静态链表.....47 | |
| 2.3.6 链式存储结构小结.....48 | |
| 2.3.7 应用举例.....48 | |
| 2.4 顺序表示与链式表示比较.....53 | |
| 2.4.1 基于空间的考虑..... 53 | |
| 2.4.2 基于时间的考虑..... 54 | |
| 2.4.3 基于语言的考虑..... 54 | |
| 2.5 习题..... 54 | |
| 第 3 章 栈和队列..... 58 | |
| 3.1 栈..... 58 | |
| 3.1.1 栈的定义..... 58 | |
| 3.1.2 栈的抽象数据类型..... 59 | |
| 3.1.3 栈的顺序存储表示与实现..... 60 | |
| 3.1.4 栈的链式存储表示与实现..... 63 | |
| 3.1.5 应用举例..... 64 | |
| 3.2 队列..... 73 | |
| 3.2.1 队列的定义..... 73 | |
| 3.2.2 队列的抽象数据类型..... 73 | |
| 3.2.3 队列的顺序存储表示 与实现..... 74 | |
| 3.2.4 队列的链式存储表示 与实现..... 78 | |
| 3.2.5 应用举例..... 80 | |
| 3.3 习题..... 83 | |
| 第 4 章 串..... 85 | |
| 4.1 串的类型定义..... 85 | |
| 4.1.1 串的定义..... 85 | |
| 4.1.2 串的抽象数据类型..... 86 | |
| 4.2 串的存储表示与实现..... 88 | |
| 4.2.1 定长顺序存储表示..... 88 | |
| 4.2.2 堆分配存储表示..... 91 | |
| 4.2.3 块链存储表示..... 95 | |
| 4.2.4 应用举例..... 96 | |
| 4.3 串的模式匹配..... 97 | |
| 4.3.1 串的模式匹配 BF 算法..... 97 | |

| | | | |
|-------------------------------|------------|-----------------------------------|------------|
| 4.3.2 串的模式匹配 KMP 算法..... | 99 | 6.3.3 线索二叉树的存储 表示与实现..... | 170 |
| 4.4 习题 | 103 | 6.3.4 应用举例 | 176 |
| 第 5 章 多维数组与广义表 | 105 | 6.4 树和森林..... | 177 |
| 5.1 多维数组 | 105 | 6.4.1 树和森林的定义..... | 177 |
| 5.1.1 数组的定义 | 105 | 6.4.2 树的抽象数据类型..... | 178 |
| 5.1.2 数组的抽象数据类型 | 106 | 6.4.3 树的存储表示..... | 179 |
| 5.1.3 数组的顺序存储 表示和实现 | 107 | 6.4.4 树和森林与二叉树的转换..... | 183 |
| 5.1.4 应用举例 | 110 | 6.4.5 树和森林的遍历..... | 187 |
| 5.2 矩阵的压缩存储..... | 111 | 6.4.6 应用举例 | 189 |
| 5.2.1 特殊矩阵 | 112 | 6.5 Huffman 树及其应用 | 197 |
| 5.2.2 稀疏矩阵 | 114 | 6.5.1 Huffman 树的定义 | 197 |
| 5.2.3 应用举例 | 124 | 6.5.2 Huffman 算法 | 199 |
| 5.3 广义表 | 128 | 6.5.3 Huffman 树的存储 表示与实现..... | 200 |
| 5.3.1 广义表的定义 | 128 | 6.5.4 Huffman 编码 | 201 |
| 5.3.2 广义表的抽象数据类型..... | 129 | 6.5.5 Huffman 编码的存储 表示与实现..... | 202 |
| 5.3.3 广义表的链式存储 表示与实现 | 130 | 6.5.6 应用举例 | 203 |
| 5.3.4 应用举例 | 138 | 6.6 习题 | 205 |
| 5.4 习题 | 142 | 第 7 章 图 | 210 |
| 第 6 章 二叉树与树 | 144 | 7.1 图的类型定义..... | 210 |
| 6.1 二叉树 | 144 | 7.1.1 图的定义 | 210 |
| 6.1.1 二叉树的定义 | 144 | 7.1.2 图的基本术语..... | 210 |
| 6.1.2 二叉树的基本术语 | 145 | 7.1.3 图的抽象数据类型..... | 214 |
| 6.1.3 二叉树的抽象数据类型..... | 147 | 7.1.4 应用举例 | 215 |
| 6.1.4 二叉树的基本性质 | 148 | 7.2 图的存储表示与实现..... | 216 |
| 6.1.5 二叉树的存储表示 | 150 | 7.2.1 邻接矩阵表示法..... | 216 |
| 6.2 遍历二叉树 | 153 | 7.2.2 邻接表表示法..... | 219 |
| 6.2.1 遍历二叉树的定义 | 154 | 7.2.3 十字链表表示法..... | 222 |
| 6.2.2 遍历二叉树的递归算法..... | 155 | 7.2.4 邻接多重表表示法..... | 224 |
| 6.2.3 遍历二叉树的非递归算法..... | 156 | 7.1.5 应用举例 | 226 |
| 6.2.4 基于遍历操作的其他算法..... | 162 | 7.3 图的遍历..... | 227 |
| 6.2.5 应用举例 | 163 | 7.3.1 深度优先搜索遍历图..... | 227 |
| 6.3 线索二叉树 | 168 | 7.3.2 广度优先搜索遍历图..... | 228 |
| 6.3.1 线索二叉树的引出 | 168 | 7.3.3 应用举例 | 229 |
| 6.3.2 线索二叉树的定义 | 169 | 7.4 最小生成树..... | 230 |

| | | | |
|-----------------------------------|-----|--------------------------|-----|
| 7.4.1 生成树 | 230 | 8.3.1 哈希表的定义 | 299 |
| 7.4.2 最小生成树 | 231 | 8.3.2 哈希函数的构造方法 | 300 |
| 7.4.3 应用举例 | 237 | 8.3.3 处理冲突的方法 | 303 |
| 7.5 最短路径 | 238 | 8.3.4 哈希表的查找和分析 | 305 |
| 7.5.1 求某个源点到其他顶点 的最段路径 | 239 | 8.3.5 应用举例 | 308 |
| 7.5.2 求每一对顶点之间的 最短路径 | 242 | 8.4 习题 | 310 |
| 7.5.3 应用举例 | 244 | 第 9 章 内部排序 | 314 |
| 7.6 拓扑排序 | 245 | 9.1 插入排序法 | 315 |
| 7.6.1 AOV 网 | 245 | 9.1.1 直接插入排序 | 315 |
| 7.6.2 拓扑排序 | 247 | 9.1.2 希尔排序 | 317 |
| 7.6.3 应用举例 | 249 | 9.1.3 应用举例 | 318 |
| 7.7 关键路径 | 250 | 9.2 交换排序法 | 319 |
| 7.7.1 AOE 网 | 250 | 9.2.1 冒泡排序 | 319 |
| 7.7.2 关键路径的概念 | 251 | 9.2.2 快速排序 | 320 |
| 7.7.3 求关键路径的算法实现 | 252 | 9.2.3 应用举例 | 323 |
| 7.7.4 应用举例 | 254 | 9.3 选择排序法 | 323 |
| 7.8 习题 | 255 | 9.3.1 直接选择排序 | 324 |
| 第 8 章 查找表 | 260 | 9.3.2 堆排序 | 325 |
| 8.1 静态查找表 | 262 | 9.3.3 应用举例 | 331 |
| 8.1.1 静态查找表的抽象 数据类型 | 262 | 9.4 归并排序法 | 331 |
| 8.1.2 静态查找表的顺序 存储表示 | 262 | 9.4.1 两个有序序列的归并 | 331 |
| 8.1.3 顺序查找 | 262 | 9.4.2 2-路归并排序 | 332 |
| 8.1.4 折半查找 | 263 | 9.4.3 应用举例 | 333 |
| 8.1.5 分块查找 | 265 | 9.5 基数排序法 | 333 |
| 8.1.6 应用举例 | 267 | 9.5.1 多关键字排序 | 333 |
| 8.2 动态查找表 | 268 | 9.5.2 链式基数排序 | 335 |
| 8.2.1 动态查找的抽象数据类型 | 268 | 9.5.3 应用举例 | 339 |
| 8.2.2 动态查找表的存储表示 | 269 | 9.6 各种内部排序法的比较 | 340 |
| 8.2.3 二叉排序树 | 269 | 9.7 习题 | 341 |
| 8.2.4 平衡二叉树 | 274 | 第 10 章 外部排序 | 344 |
| 8.2.5 B 树和 B ⁺ 树 | 280 | 10.1 外存储设备简介 | 344 |
| 8.2.6 键树 | 288 | 10.1.1 磁带信息的存取 | 344 |
| 8.2.7 应用举例 | 292 | 10.1.2 磁盘信息的存取 | 345 |
| 8.3 哈希表 | 298 | 10.1.3 光盘信息的存取 | 346 |
| | | 10.2 磁带文件归并排序 | 347 |
| | | 10.2.1 平衡归并排序 | 347 |
| | | 10.2.2 多步归并排序 | 348 |

| | | | |
|------------------------|------------|--------------------------------|------------|
| 10.2.3 应用举例 | 353 | 11.4 ISAM 和 VSAM 文件 | 371 |
| 10.3 磁盘文件归并排序 | 356 | 11.4.1 ISAM 文件 | 371 |
| 10.3.1 初始归并段的生成 | 356 | 11.4.2 VSAM 文件 | 374 |
| 10.3.2 置换选择排序法 | 357 | 11.5 哈希文件 | 376 |
| 10.3.3 应用举例 | 358 | 11.5.1 哈希文件的操作 | 377 |
| 10.4 最佳归并树 | 359 | 11.5.2 哈希文件的特点 | 378 |
| 10.4.1 最佳归并树的定义 | 359 | 11.6 多关键字文件 | 378 |
| 10.4.2 最佳归并树的设计 | 361 | 11.6.1 多重表文件 | 378 |
| 10.4.3 应用举例 | 362 | 11.6.2 倒排文件 | 380 |
| 10.5 习题 | 362 | 11.7 应用举例 | 381 |
| 第 11 章 文件 | 364 | 11.8 习题 | 384 |
| 11.1 基本概念 | 364 | 第 12 章 数据结构程序设计方法 | 386 |
| 11.1.1 文件的概念 | 364 | 12.1 从问题到程序的求解过程 | 386 |
| 11.1.2 文件的分类 | 365 | 12.1.1 建立数据结构模型设计 | |
| 11.1.3 文件的逻辑结构 | 365 | 抽象数据类型 | 386 |
| 11.1.4 文件的物理结构 | 366 | 12.1.2 算法设计 | 387 |
| 11.2 顺序文件 | 366 | 12.1.3 实现抽象数据类型 | 388 |
| 11.2.1 顺序文件的查找 | 367 | 12.1.4 编制程序代码并进行 | |
| 11.2.2 顺序文件的修改 | 367 | 静态测试和动态调试 | 389 |
| 11.2.3 顺序文件的特点 | 368 | 12.2 程序的规范说明 | 391 |
| 11.3 索引文件 | 368 | 12.3 应用举例 | 392 |
| 11.3.1 索引文件的分类 | 368 | 附录 A 部分习题答案 | 402 |
| 11.3.2 索引文件的存储 | 369 | 参考书目 | 409 |
| 11.3.3 索引文件的操作 | 369 | | |
| 11.3.4 利用查找表建立 | | | |
| 多级索引 | 370 | | |

第 1 章 绪 论

学习要点

- 了解数据结构的含义及其相关概念，特别是数据结构的两个类型和 4 种基本存储方式
- 熟悉类 C 语言描述算法的方法
- 明确算法的 5 个重要特性及 4 个设计要求
- 掌握估算算法的时间和空间复杂度的方法

1.1 数据结构的讨论范畴

众所周知，20 世纪 40 年代，电子计算机问世的直接原因是解决弹道学的计算问题。早期的电子计算机，其应用范围只局限于科学和工程的计算，处理的对象是纯数值性的数据，通常人们把这类问题称为数值计算。

半个多世纪以来，计算机科学、计算机软件及计算机硬件技术的发展异常迅猛，这不仅表现在计算机的运算速度不断提高，信息存储量日益增大，硬件价格逐步下降，更重要的是其应用已经远远超出了科学计算的范围，广泛地应用于情报检索、企业管理、系统工程，乃至人类社会活动的一切领域。与此同时，计算机的处理对象也从简单的纯数值性数据发展到非数值性的并具有一定结构的数据，如文本、图形、图像、音频、视频及动画等。

由于数据的组织形式和表示方式直接关系到计算机对数据的处理效率，而计算机系统程序和许多应用程序的规模很大，结构非常复杂，处理对象又多为非数值性数据，因此单凭程序设计人员的经验和技巧难以保证设计出高效可靠的程序。为了更好地进行程序设计，有效地使用计算机，就需要对计算机程序加工处理的对象进行系统和深入地研究。研究各种数据的特性以及数据之间存在的关系，进而根据实际应用的要求，合理地组织和存储数据，设计出相应的算法，这就是“数据结构”要讨论的问题。

为了增强对数据结构的感性认识，下面举例说明。

例 1.1 学生情况登记管理。

表 1.1 中的学生情况登记表是一个数据结构。表中每个学生的情况为一个记录，它由姓名、性别、年龄、籍贯、班别和成绩 6 个数据项组成。其中，姓名、性别、年龄、籍贯、班别为原子项；成绩为组合项，它又分为数学、物理、化学、外语 4 个原子项。

学生情况登记管理的主要功能包括查询、浏览、插入、修改、删除和统计等。

在这种数据结构中，计算机处理的数据之间存在的是一种“一个对一个”的简单线性关系，称为线性数据结构。

表 1.1 学生情况登记表

| 姓名 | 性别 | 年龄 | 籍贯 | 班别 | 成绩 | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | | | | | 数学 | 物理 | 化学 | 外语 |
| 张三 | 女 | 18 | 北京 | 6003 | 80 | 70 | 90 | 90 |
| 李四 | 男 | 20 | 上海 | 6002 | 90 | 95 | 85 | 80 |
| 王五 | 男 | 19 | 长沙 | 6004 | 85 | 90 | 80 | 75 |
| 赵六 | 女 | 19 | 湖北 | 6001 | 85 | 80 | 75 | 90 |
| | | | | | | | | |

例 1.2 Internet 域名查询系统。

图 1.1 中的数据结构像一棵根在上、枝在下的倒挂的树，清晰地描述了 Internet 域名和站点的 IP 地址之间的关系。输入某站点的域名，在 Internet 域名系统的树型结构中进行搜索，直至域名全部匹配成功或匹配失败：若成功则给出该站点的 IP 地址，否则给出找不到该站点的信息。搜索的过程就是从树根沿树枝到某个叶子的过程。

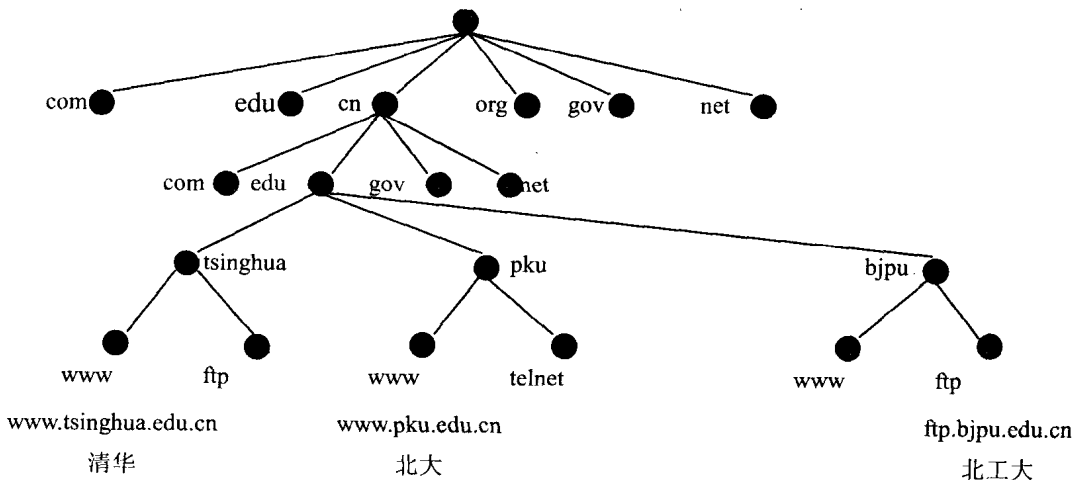


图 1.1 Internet 域名查询树

在这种数据结构中，计算机处理的数据之间存在的是一种“一个对多个”的层次关系，称为树型数据结构。

例 1.3 建立最小造价通信网络。

在 n 个城市之间建立通信网络，要求在其中任意两个城市之间都有直接的或间接的通信线路，在已知某些城市之间直接通信线路预算造价的情况下，使网络的造价最低。

当 n 很大时，这样的问题只能用计算机来求解。我们用图 1.2(a) 来描述 7 个城市之间的通信线路，其中：图中圆圈表示一个城市，两个圆圈之间的连线表示对应城市之间的通信线路，连线上的数值表示该通信线路的造价。这一描述的结构为图状结构，利用计算机可以求出满足要求的通信网络，如图 1.2(b) 所示。

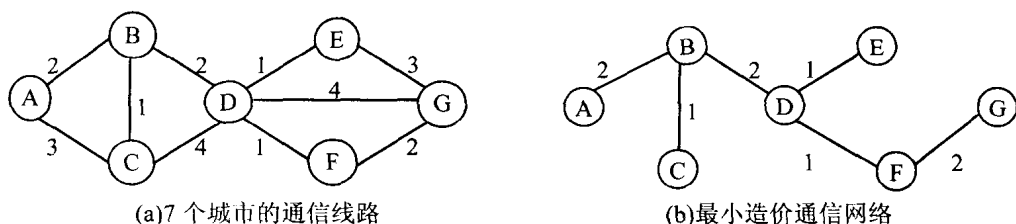


图 1.2 建立最小造价通信网络

通过上面 3 个例子可以看出：数据结构中数据元素之间存在着逻辑关系。数据结构就是解决如何分析数据元素之间的关系、如何确立合适的逻辑结构、如何存储这些数据，并对为完成数据操作所设计的算法做出时间和空间的分析。

简单来说，数据结构是研究非数值计算的程序设计问题中数据以及它们之间的逻辑关系和对数据的操作的一门课程。重点分析数据之间的抽象的相互关系，而不涉及数据的具体内容。

瑞士著名计算机科学家，Pascal 语言的发明者 Niklaus Wirth 早在 1976 年出版了一本书，名为《算法 + 数据结构 = 程序设计》，此书名很快就成为计算机工作者之间流传的一句名言。至今，尽管新的技术方法不断涌现，这句名言依然焕发着无限的生命力，它借助面向对象知识的普及，使数据结构技术更加完善和易于使用。由此，也说明了数据结构在计算机学科中的地位 and 不可替代的独特作用。

1.2 数据结构的发展概况

回顾一下数据结构课程的发展与形成过程对于理解数据结构的内容和重要性是很有意义的。

(1) 20 世纪 60 年代初期，各种高级程序设计语言相继出现。

相继出现的这些高级语言所能描述的数据类型逐渐丰富。例如，FORTRAN 语言允许使用复数类型和数组类型，COBOL 语言和 PL/1 语言允许使用字符类型和记录类型。特别是数组类型和记录类型都已经具备“结构”的雏型，其中数组是把一批相关联的同类型的数据看作一个整体，记录是把一组相关联的不同类型的数据看作一个整体。有了结构的概念之后，各个数据之间就不再是孤立的，而是相互联系的，这样便于表示和存储，也便于处理。

例如，一个银行储户的基本情况包括姓名、地址、身份证号、开户日期和存款余额等项目的内容，把这些数据项组织起来就构成了一个储户的基本资料，称为该储户的记录；把各储户的记录组合起来就是一个数组；把银行全体储户的数据按照它们之间的内在关系组织在一起，这就是一个数据结构。

这期间，数据结构的内容已经散见于《编译原理》和《操作系统》等课程之中。

(2) 20 世纪 60 年代中期，开设表处理语言课程。

有些国家的大学在这期间开始设立有关“数据结构”课程，但是当时课程的名称并不叫数据结构，而命名为表处理语言。它的主要内容是研究当时已经出现的几种典型的表处

理语言:

- J.Weizenbaum 在 20 世纪 50 年代初设计的 SLIP 语言(简单表处理语言);
- A.Newell 等人在 1954 年~1959 年设计的 IPL-V 语言(信息处理系统);
- J.McCarthy 在 1959 年~1960 年设计的 LISP 语言(表处理语言);
- D.Farber 等人在 1962 年设计的 SNOBOL 语言(串处理语言)。

它们的共同特点是数据对象的结构形式是表结构或树结构。例如 LISP 的数据结构就是二叉树结构, SNOBOL 的数据结构是表和树结构。

上述的几种语言是以数据为中心, 为处理非数值问题设计的, 例如 LISP 就是为解决人工智能问题而设计的表处理语言。而我们所熟知的 FORTRAN、ALGOL 等算法语言则是为解决数值问题而设计的, 它们侧重于以程序为中心。例如 ALGOL-60 分程序和过程就是以程序为中心的例子。

以程序为中心的观点侧重于建立程序, 只是当数据成为程序加工的对象时, 才考虑到数据。这种观点适合于数值计算问题, 这类问题属于在简单数据结构上进行复杂函数变换的问题。以数据为中心的观点是把数据结构作为问题的中心部分(如数据库), 而把程序看成是围绕着数据结构缓慢爬行的小虫, 它时而询问, 时而修改或扩充当前驻留在内存中的数据。这种观点适合于航空订票系统、信息管理系统、情报检索系统等非数值问题的解决, 它们都要求采用复杂的数据结构描述系统的状态, 它们的运算是实现对于数据结构的访问或改变等。某些科学家曾断言, 程序设计以数据为中心的观点, 将对未来程序设计语言的设计产生重大影响。

(3) 20 世纪 60 年代后期, 美国计算机界首先使用数据结构这一概念。

1968 年, 美国的部分大学计算机科学系的教学计划中明确规定《数据结构》为一门独立课程, 但是对该课程的内容范围并没有做具体的限定。最初, 数据结构几乎和图论是同义语, 特别是表和树的理论(这些理论是描述分层数据的有利武器)。随后, 这个概念又扩充到包括网络、代数、集合论、关系等现在称之为“离散数学结构”的内容。由于数据必须在计算机中进行处理, 因此不能局限于数据本身的数学概念的研究, 还必须考虑数据的物理结构, 即数据在存储器中的表示问题, 这就进一步扩大了数据结构的内容。

美国著名计算机科学教授 D.E.Knuth(唐·欧·克努特)所著的 *The Art of Computer Programming Volume 1*(《计算机程序设计技巧》第一卷)的出版, 对计算机科学的发展做出了重大贡献。该书论证了任何语言都可以采用像表处理语言的技术。这部著作全面系统地阐述了数据的逻辑结构和存储结构, 并给出了各种典型的算法, 为数据结构奠定了基础。

(4) 20 世纪 70 年代初期, 提出“程序设计 = 算法 + 数据结构”。

随着大型程序及大规模文件系统的出现, 使程序设计方法和数据结构已经不再是理论课题, 它们在实际中受到了极大的重视。人们认为, 程序设计的本质是对要处理的问题选择一种好的数据结构, 同时在此结构上施加一种好的算法。1976 年瑞士著名计算机科学家 Niklaus Wirth 出版的为《算法 + 数据结构 = 程序设计》一书就是这种程序设计思想的著名代表之一。

(5) 20 世纪 70 年代中期, 数据结构逐步成为一门比较完整的学科。

随着数据库系统和情报检索系统的不断发展, 在数据结构的技术中又增加了文件管理(特别是大型文件的组织等)的内容和 B-树、B⁺树的知识, 数据结构逐步成为了一门比较

完整的学科。

(6) 20世纪80年代初, 在我国的大学中开设数据结构课程。

在现行的计算机专业教学计划中, 已经把数据结构课程列为核心课程之一。许多非计算机专业也把数据结构作为必修和选修课程。数据结构是一门是介于数学、计算机硬件和计算机软件三者之间的计算机专业基础课, 是程序设计方法学、数据库系统、操作系统、编译原理、软件工程学等课程的先修课程。在实际应用中, 它是设计和实现大型应用软件的重要基础。

值得注意的是, 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, 从抽象数据类型的观点来讨论数据结构, 已经成为一种新的趋势, 越来越被人们所重视。

1.3 数据结构的相关概念

在这一节中, 将对与数据结构相关的、并且在以后各章中反复出现的一些概念和术语给出明确的含义。

1.3.1 基本概念和术语

1. 数据(data)

数据是对客观事物的符号表示, 在计算机科学中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”。

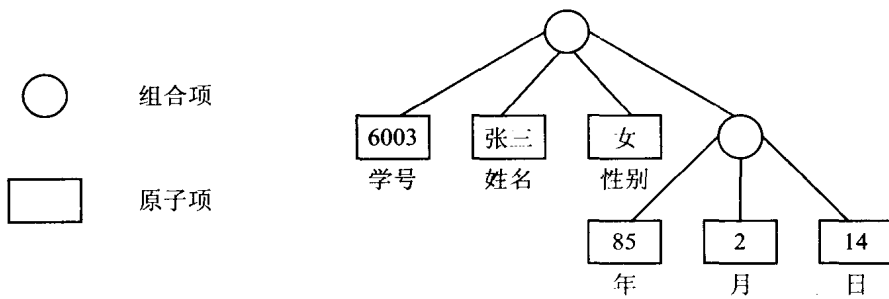
例如, 一个利用数值分析方法解代数方程的程序, 其处理对象是整数和实数; 一个编译程序或文字处理程序的处理对象是字符串。因此, 对计算机科学而言, 数据的含义极为广泛, 图形、图像、色彩和声音等都可以通过编码而归之于数据的范畴。

2. 数据元素(data element)

数据元素是数据的基本单位, 在计算机程序中通常作为一个整体进行考虑和处理。有时, 一个数据元素可以是不可分割的原子(atom), 也可以由若干个数据项(data item)组成。如果一个数据项由若干个数据项组成(如出生日期), 则称为组合项; 如果一个数据项由一个数据项组成(如性别), 则称为原子项。

例如, 某程序处理的数据是学生情况登记表, 每个学生的信息就是一个数据元素, 其中的学生信息中的每一项(比如学号、姓名、性别、出生年月等)是这个数据元素中的数据项。图1.3所示即为上述数据元素的内部结构。

数据元素的同义词有: 记录(record)、节点(node)和顶点(vertex)。它们的名称虽然不同, 但所表示的意义却是一样的。通常, 在顺序结构中多用“元素”, 在链式结构中多用“节点”, 在图和文件中又分别使用“顶点”和“记录”。



3. 数据对象(data object)

数据对象是性质相同的数据元素的集合，它是数据的一个子集。

例如，在电话号码查询系统中，数据对象是全体的电话用户；在银行业务处理系统中，数据对象是全体储户的资料及全体贷款客户的资料。

4. 数据关系(data relation)

在数据对象中各数据元素之间存在着某种关系，这种关系反映了数据对象中数据元素所固有的一种结构。在数据处理领域，通常把数据之间这种固有的关系简单地用前驱和后继来描述。

例如，在编写家庭族谱时，数据对象是家庭中的所有成员，对家族中某成员的描述就是一个数据元素，各数据元素之间存在着血缘关系；父亲是儿子的前驱，儿子是父亲的后继，小孩子没有后继。又如，一张按名次排列的成绩表，数据对象是全班同学，对某个同学属性(姓名、成绩等)的描述就是一个数据元素，各数据元素之间存在着名次关系；第一名没有前驱，其后继是第二名；第二名的前驱是第一名，其后继是第三名。

由此，前驱和后继关系所表示的实际意义随着数据对象的不同而不同。一般来说，数据元素之间的任何关系都可以用前驱和后继关系来描述。

5. 关键字(key)

关键字(关键码)指的是数据元素中能够起标识作用的数据项。其中能够起惟一标识作用的关键字称为“主关键字”(main key)，反之称为“次关键字”(secondary key)。

例如，在学生情况登记中，学号和姓名为关键字，其中学号为主关键字，姓名为次关键字。

1.3.2 数据结构

对于数据结构这个概念，至今尚未有一个公认的定义，不同的人在使用这个词时所表达的意思有所不同。本书将给出一个通用的描述。

1. 数据结构(data structure)

数据结构是带有结构特性的数据元素的集合，它研究的是数据的逻辑结构和数据的物

理结构以及它们之间的相互关系,并对这种结构定义相适应的运算,设计出相应的算法,并确保经过这些运算以后所得到的新结构仍然是原来的结构类型。

数据的逻辑结构和物理结构是数据结构的两个密切相关的方面,同一逻辑结构可以对不同的存储结构。算法的设计取决于数据的逻辑结构,而算法的实现依赖于指定的存储结构。

2. 数据的逻辑结构(data logical structure)

数据的逻辑结构是对数据元素之间的逻辑(数学)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。

数据的逻辑结构定义是对操作对象的一种数学描述,换句话说,是从操作对象抽象出来的数学模型,不依赖于计算机。下面用数学方法给出数据的逻辑结构定义。

设 D 表示数据元素的集合, R 表示 D 上关系的集合(即 R 反映了 D 中各元素的前驱和后继关系),则一个数据的逻辑结构 B 可以表示为: $B = \{D, R\}$ 。

根据数据元素之间的关系的不同特性,通常有 4 类基本结构。

(1) 集合(set): 在该数据结构中,只有数据元素,它们之间除了“同属于一个集合”外别无其他的关系,即 $R = \{\}$ 。集合是数据结构的一种特例,如图 1.4 (a) 所示。

(2) 线性结构(line structure): 在该数据结构中,除了第 1 个数据元素外,其他各元素有惟一的前驱;除了最后一个数据元素外,其他各元素有惟一的后继。数据结构中的数据元素之间存在一个对一的关系,如图 1.4 (b) 所示。

(3) 树型结构(tree structure): 在该数据结构中,除了一个根数据元素(节点)外,其他各元素(节点)有惟一的前驱;所有数据元素(节点)都可以有多个后继。数据结构中的数据元素之间存在一个对多个的关系,如图 1.4 (c) 所示。

(4) 图状结构或网状结构(graph or net structure): 在该数据结构中,各数据元素可以有多个前驱或后继。数据结构中的数据元素之间存在多个对多个的关系,如图 1.4(d)所示。

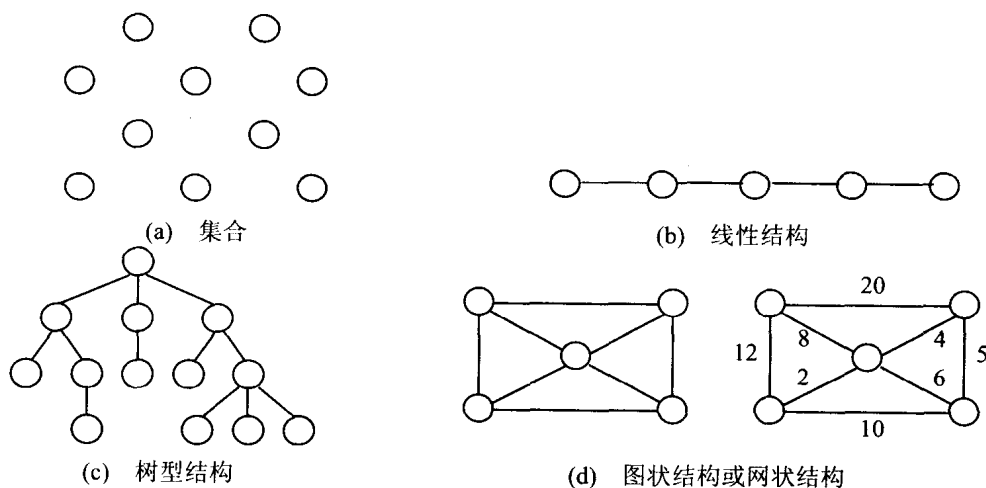


图 1.4 4 类基本逻辑结构的示意图

一般把树型结构和图状结构称为非线性结构。