

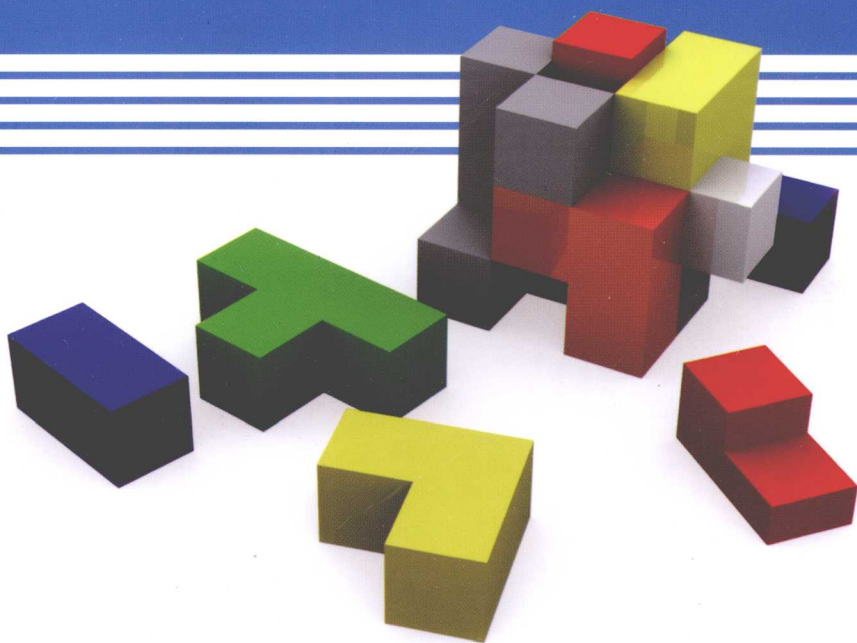


高等学校数据结构课程系列教材

数据结构教程

(C# 语言描述)

李春葆 主编



清华大学出版社

013025957

TP311.12-43

169

高等学校数据结构课程系列教材

数据结构教程(C#语言描述)

李春葆 尹为民 喻丹丹 李春葆 主编
蒋晶珏 安杨 编著



清华大学出版社
北京

TP311.12-43
169



北航

C1632914

内 容 简 介

本书系统地介绍了常用的数据结构以及排序、查找的各种算法,阐述了各种数据结构的逻辑关系、存储表示及运算操作,并采用C#语言描述数据组织和算法实现。

本书既注重原理,又注重实践,配有大量图表、实践教学项目和习题,内容丰富,概念讲解清楚,表达严谨,逻辑性强,语言精练,可读性好。

本书既便于教师课堂讲授,又便于自学者阅读,可作为高等院校计算机相关专业本科生、专科生的教材,也可供广大从事计算机应用的科技人员参考。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构教程(C#语言描述)/李春葆主编. —北京:清华大学出版社,2013.2

(高等学校数据结构课程系列教材)

ISBN 978-7-302-30517-0

I. ①数… II. ①李… III. ①数据结构—教材 ②C语言—程序设计—教材 IV. ①TP311.12
②TP312

中国版本图书馆CIP数据核字(2012)第257453号

责任编辑:魏江江 王冰飞

封面设计:杨 兮

责任校对:梁 毅

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:24

字 数:596千字

版 次:2013年2月第1版

印 次:2013年2月第1次印刷

印 数:1~3000

定 价:39.00元

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大,社会对计算机专业人才的需求不仅体现在数量的增加上,而且体现在质量要求的提高上,培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前,我国共有16个国家重点学科、20个博士点一级学科、28个博士点二级学科集中在教育部部属重点大学,这些高校在计算机教学和科研方面具有一定优势,并且大多以国际著名大学计算机教育为参照系,具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系,形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础,其中专业教材的使用和建设则是这种氛围的重要组成部分,一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势,特别是专业教材建设上的优势,同时配合各重点大学的计算机学科建设和专业课程教学需要,在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下,清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”,同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

1. 面向学科发展的前沿,适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础,反映基本理论和原理的综合应用,重视实践和应用环节。

2. 反映教学需要,促进教学发展。教材要能适应多样化的教学需要,正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

3. 实施精品战略,突出重点,保证质量。规划教材建设的重点依然是专业基础课和专业主干课;特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现重点大学

计算机专业教学内容和课程体系改革成果的教材。

4. 主张一纲多本,合理配套。专业基础课和专业主干课教材要配套,同一门课程可以有多个具有不同内容特点的教材。处理好教材统一性与多样化的关系;基本教材与辅助教材以及教学参考书的关系;文字教材与软件教材的关系,实现教材系列资源配套。

5. 依靠专家,择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

前言

计算机是数据处理的工具。数据结构课程作为计算机科学与技术专业的核心课程,主要讲授数据的各种组织方式以及建立在这些结构之上的各种运算算法的实现,它不仅为计算机语言程序设计提供了方法性的指导,还在一个更高的层次上总结了程序设计的常用方法和常用技巧。

要学好数据结构课程,首先要从宏观上理解本课程的目的和地位。该课程是在学完某种程序设计语言后开设的。仅掌握一门程序设计语言,不一定能编写出“好程序”,数据结构课程就是为写好程序服务的。程序是用于数据计算的,在编写程序之前要理解数据的结构,这里是指数据的逻辑结构。从数据元素之间的相邻关系归纳起来,数据的逻辑结构主要有线性结构、树形结构和图形结构。仅弄清数据逻辑结构是不够的,还要将这些数据存放到计算机内,这称为数据的存储结构。一种数据逻辑结构可能由多种存储结构实现。当设计好数据存储结构后,就可以对其进行操作了,这种操作的指令序列称为算法。不同的功能对应不同的算法,同一功能也可以有多种算法,从算法的时间和空间来分析,可以得到最佳算法。数据结构课程总结和归纳了在软件开发中常用的数据结构,从数据逻辑结构、存储结构和基本运算算法设计3个层面来讨论,可以提高学生基本的数据组织和处理能力,为后续课程,如操作系统、数据库原理和编译原理等的学习打下基础。

本书是作者根据近20年的教学经验,参考近年国内外出版的多种数据结构以及C#面向对象程序设计教材,考虑教与学的特点,合理地进行内容的取舍,精心组织编写而成的。目前,国内外数据结构教材大多数采用C/C++语言描述算法,考虑到C#语言与C/C++的一致性和良好的Windows界面设计优点,本书采用C#语言面向对象方法描述算法和实验程序设计。

全书由10章构成,各章内容如下:

第1章绪论,介绍数据结构的基本概念、采用C#语言描述算法的特点、算法分析方法和如何设计好算法等。

第2章线性表,介绍线性表的定义、线性表的两种主要的存储结构和各种基本运算算法设计,最后通过示例讨论线性表的应用。

第3章栈和队列,介绍栈的定义、栈的存储结构、栈的各种基本运算算法设计和栈的应用;队列的定义、队列的存储结构和队列的各种基本运算算法设计和队列的应用。

第4章串,介绍串的定义、串的存储结构、串的各种基本运算算法设计和串的模式匹配算法。

第5章数组和广义表,介绍数组的定义、几种特殊矩阵的压缩存储方式、稀疏矩阵的压缩存储及相关算法设计;递归的定义和递归算法设计方法;广义表的定义、广义表的存储结构及相关算法设计方法。

第6章树和二叉树,介绍树的定义、树的逻辑表示方法、树的性质、树的遍历和树的存储结构二叉树;介绍二叉树的定义、二叉树的性质、树/森林和二叉树的转换与还原、二叉树存储结构、二叉树基本运算算法设计、二叉树的递归和非递归遍历算法、二叉树的构造、线索二叉树和哈夫曼树等。

第7章图,介绍图的定义、图的存储结构、图的基本运算算法设计、图的两种遍历算法以及图的应用(包括图的最小生成树、最短路径、拓扑排序和关键路径等)。

第8章查找,介绍查找的定义、线性表上的各种查找方法、树表上的各种查找方法以及哈希表查找方法等。

第9章内排序,介绍排序的定义、插入排序方法、交换排序方法、选择排序方法、归并排序方法和基数排序方法,并对各种排序方法进行了比较。

第10章外排序,介绍外排序的定义、外排序的基本步骤,重点讨论了磁盘排序中的各种算法。

另外,附录A中给出各章练习题单项选择题部分的答案。

本书主要特点如下:

(1) 结构清晰,内容丰富,文字叙述简洁明了,可读性强。

(2) 图文并茂,全书用300多幅图来表述和讲解数据的组织结构和算法设计思想。

(3) 力求归纳各类算法设计的规律,如单链表算法中很多是基于建表算法的,二叉树算法中很多是基于遍历算法的,图算法中很多是基于深度优先遍历的。如果读者掌握了建表算法、二叉树的遍历算法和图遍历算法,那么设计相关算法就会驾轻就熟了。

(4) 深入讨论递归算法设计的方法。递归算法设计是数据结构课程中的难点之一。作者从递归模型入手,介绍了从求解问题中提取递归模型的通用方法,讲解了从递归模型到递归算法设计的基本规律。

(5) 书中含有大量的实践项目,全面覆盖并超越了教育部制定的《高等学校计算机科学与技术专业实践教学体系与规范》中数据结构课程的实践教学要求。作者已在Visual Studio.NET 2005/2008开发环境中全部调试并通过这些实践项目。

本书的编写工作得到湖北省教育厅和武汉大学教学研究项目《计算机科学与技术专业课程体系改革》的大力支持,特别是国家级名师何炎祥教授和主管教学工作的王丽娜副院长对本书的编写给予了建设性的指导,国家珠峰计划——武汉大学计算机弘毅班的两届学生

和众多编者授课的本科生提出了许多富有启发的建议,清华大学出版社魏江江主任全力支持本书的编写工作,作者在此一并表示衷心感谢!

本书是课程组全体教师多年教学经验的总结和体现,尽管作者不遗余力,但由于水平所限,仍难免有错误和不足之处,敬请广大教师和同学们批评指正。欢迎读者通过 licb1964@126.com 邮箱跟作者联系,在此表示万分感谢!

编 者

2012年10月

| | |
|-----------------------------|----|
| 第 1 章 绪论 | 1 |
| 1.1 什么是数据结构 | 1 |
| 1.1.1 数据结构的定义 | 1 |
| 1.1.2 数据的逻辑结构 | 2 |
| 1.1.3 数据的存储结构 | 5 |
| 1.1.4 数据的运算 | 8 |
| 1.1.5 数据结构和数据类型 | 10 |
| 1.2 算法及其描述 | 15 |
| 1.2.1 什么是算法 | 15 |
| 1.2.2 算法描述 | 16 |
| 1.3 算法分析 | 18 |
| 1.3.1 算法的特性和算法设计的目标 | 18 |
| 1.3.2 算法时间效率分析 | 19 |
| 1.3.3 算法存储空间分析 | 22 |
| 1.4 数据结构的目标 | 22 |
| 本章小结 | 23 |
| 练习题 1 | 23 |
| 第 2 章 线性表 | 26 |
| 2.1 线性表的定义 | 26 |
| 2.1.1 什么是线性表 | 26 |
| 2.1.2 线性表的抽象数据类型描述 | 27 |
| 2.2 线性表的顺序存储结构 | 27 |
| 2.2.1 线性表的顺序存储结构——顺序表 | 27 |
| 2.2.2 顺序表基本运算的实现 | 28 |
| 2.3 线性表的链式存储结构 | 36 |
| 2.3.1 线性表的链式存储结构——链表 | 36 |

| | | |
|--------------|--------------------------|------------|
| 2.3.2 | 单链表 | 37 |
| 2.3.3 | 双链表 | 47 |
| 2.3.4 | 循环链表 | 52 |
| 2.4 | 线性表的应用 | 58 |
| | 本章小结 | 65 |
| | 练习题 2 | 66 |
| 第 3 章 | 栈和队列 | 70 |
| 3.1 | 栈 | 70 |
| 3.1.1 | 栈的定义 | 70 |
| 3.1.2 | 栈的顺序存储结构及其基本运算的实现 | 71 |
| 3.1.3 | 栈的链式存储结构及其基本运算的实现 | 75 |
| 3.1.4 | 栈的应用 | 78 |
| 3.2 | 队列 | 88 |
| 3.2.1 | 队列的定义 | 89 |
| 3.2.2 | 队列的顺序存储结构及其基本运算的实现 | 90 |
| 3.2.3 | 队列的链式存储结构及其基本运算的实现 | 95 |
| 3.2.4 | 队列的应用 | 99 |
| | 本章小结 | 102 |
| | 练习题 3 | 102 |
| 第 4 章 | 串 | 106 |
| 4.1 | 串的基本概念 | 106 |
| 4.1.1 | 什么是串 | 106 |
| 4.1.2 | 串的抽象数据类型 | 107 |
| 4.2 | 串的存储结构 | 108 |
| 4.2.1 | 串的顺序存储结构——顺序串 | 108 |
| 4.2.2 | 串的链式存储结构——链串 | 113 |
| 4.3 | 串的模式匹配 | 120 |
| 4.3.1 | Brute-Force 算法 | 121 |
| 4.3.2 | KMP 算法 | 123 |
| | 本章小结 | 129 |
| | 练习题 4 | 130 |
| 第 5 章 | 数组和广义表 | 131 |
| 5.1 | 数组 | 131 |
| 5.1.1 | 数组的定义 | 131 |
| 5.1.2 | 数组的存储结构 | 132 |
| 5.1.3 | 特殊矩阵的压缩存储 | 134 |

| | | |
|--------------|---------------------|------------|
| 5.2 | 稀疏矩阵 | 137 |
| 5.2.1 | 稀疏矩阵的三元组表示 | 137 |
| 5.2.2 | 稀疏矩阵的十字链表表示 | 141 |
| 5.3 | 递归 | 144 |
| 5.3.1 | 递归的定义 | 144 |
| 5.3.2 | 何时使用递归 | 145 |
| 5.3.3 | 递归模型 | 146 |
| 5.3.4 | 递归算法设计的步骤 | 147 |
| 5.3.5 | 递归算法转换为非递归算法 | 149 |
| 5.4 | 广义表 | 152 |
| 5.4.1 | 广义表的定义 | 152 |
| 5.4.2 | 广义表的存储结构 | 154 |
| 5.4.3 | 广义表的运算 | 155 |
| | 本章小结 | 161 |
| | 练习题 5 | 161 |
| 第 6 章 | 树和二叉树 | 163 |
| 6.1 | 树 | 163 |
| 6.1.1 | 树的定义 | 163 |
| 6.1.2 | 树的逻辑结构表示方法 | 164 |
| 6.1.3 | 树的基本术语 | 165 |
| 6.1.4 | 树的性质 | 166 |
| 6.1.5 | 树的基本运算 | 167 |
| 6.1.6 | 树的存储结构 | 168 |
| 6.2 | 二叉树 | 170 |
| 6.2.1 | 二叉树的定义 | 171 |
| 6.2.2 | 二叉树的性质 | 172 |
| 6.2.3 | 二叉树与树、森林之间的转换 | 174 |
| 6.2.4 | 二叉树的存储结构 | 177 |
| 6.2.5 | 二叉树的基本运算及其实现 | 179 |
| 6.2.6 | 二叉树的遍历 | 182 |
| 6.2.7 | 二叉树的构造 | 200 |
| 6.2.8 | 线索二叉树 | 205 |
| 6.3 | 哈夫曼树 | 210 |
| 6.3.1 | 哈夫曼树的定义 | 210 |
| 6.3.2 | 哈夫曼树的构造算法 | 211 |
| 6.3.3 | 哈夫曼编码 | 213 |
| | 本章小结 | 216 |
| | 练习题 6 | 216 |

| | |
|--------------------------|-----|
| 第7章 图 | 220 |
| 7.1 图的基本概念 | 220 |
| 7.1.1 图的定义..... | 220 |
| 7.1.2 图的基本术语..... | 221 |
| 7.2 图的存储结构和基本运算的实现 | 223 |
| 7.2.1 邻接矩阵存储方法..... | 224 |
| 7.2.2 邻接表存储方法..... | 225 |
| 7.3 图的遍历 | 230 |
| 7.3.1 图的遍历的概念..... | 231 |
| 7.3.2 深度优先遍历..... | 231 |
| 7.3.3 广度优先遍历..... | 233 |
| 7.3.4 非连通图的遍历..... | 235 |
| 7.3.5 图遍历算法的应用..... | 236 |
| 7.4 图的应用 | 244 |
| 7.4.1 生成树和最小生成树..... | 244 |
| 7.4.2 最短路径..... | 252 |
| 7.4.3 拓扑排序..... | 260 |
| 7.4.4 AOE网与关键路径 | 262 |
| 本章小结..... | 269 |
| 练习题7 | 269 |
| 第8章 查找 | 274 |
| 8.1 查找的基本概念 | 274 |
| 8.2 线性表的查找 | 275 |
| 8.2.1 顺序查找..... | 276 |
| 8.2.2 折半查找..... | 277 |
| 8.2.3 索引存储结构和分块查找..... | 280 |
| 8.3 树表的查找 | 285 |
| 8.3.1 二叉排序树..... | 285 |
| 8.3.2 平衡二叉树..... | 293 |
| 8.3.3 B-树 | 299 |
| 8.3.4 B+树 | 304 |
| 8.4 哈希表查找 | 306 |
| 8.4.1 哈希表的基本概念..... | 306 |
| 8.4.2 哈希函数构造方法..... | 306 |
| 8.4.3 哈希冲突的解决方法..... | 307 |
| 8.4.4 哈希表查找及性能分析..... | 310 |
| 本章小结..... | 314 |

| | |
|------------------------------|------------|
| 练习题 8 | 314 |
| 第 9 章 内排序 | 317 |
| 9.1 排序的基本概念 | 317 |
| 9.2 插入排序 | 319 |
| 9.2.1 直接插入排序 | 319 |
| 9.2.2 折半插入排序 | 322 |
| 9.2.3 希尔排序 | 323 |
| 9.3 交换排序 | 325 |
| 9.3.1 冒泡排序 | 325 |
| 9.3.2 快速排序 | 327 |
| 9.4 选择排序 | 330 |
| 9.4.1 简单选择排序 | 331 |
| 9.4.2 堆排序 | 332 |
| 9.5 归并排序 | 336 |
| 9.6 基数排序 | 338 |
| 9.7 各种内排序方法的比较和选择 | 342 |
| 本章小结 | 350 |
| 练习题 9 | 350 |
| 第 10 章 外排序 | 353 |
| 10.1 外排序概述 | 353 |
| 10.2 磁盘排序 | 354 |
| 10.2.1 磁盘排序过程 | 354 |
| 10.2.2 生成初始归并段 | 355 |
| 10.2.3 多路平衡归并 | 357 |
| 10.2.4 最佳归并树 | 360 |
| 本章小结 | 364 |
| 练习题 10 | 364 |
| 附录 A 部分练习题参考答案 | 366 |
| 参考文献 | 368 |

数据结构作为一门独立的课程,最早在美国的一些大学开设。1968年,美国 Donald E. Knuth 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》系统地阐述了数据的逻辑结构和存储结构及其操作,是数据结构的经典之作。20世纪60年代末出现了大型程序,结构化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,即“程序=数据结构+算法”。从20世纪70年代开始,数据结构得到了迅速发展,编译程序、操作系统和数据库管理系统等都涉及到数据元素的组织以及在存储器中的分配,数据结构技术成为设计和实现大型系统软件和应用软件的关键技术。

数据结构课程通过介绍一些典型数据结构的特性来讨论基本的数据组织和数据处理方法。本课程可以使学生对数据结构的逻辑结构和存储结构具有明确的基本概念和必要的基础知识,对定义在数据结构上的基本运算有较强的理解能力,学会分析研究计算机加工的数据结构的特性,以便为应用涉及的数据选择适当的逻辑结构和存储结构,并能设计出较高质量的算法。

1.1 什么是数据结构

在了解数据结构的重要性之后,开始讨论数据结构的定义。本节先从一个简单的学生表例子入手,继而给出数据结构的严格定义,接着分析数据结构的几种类型,最后给出数据结构和数据类型之间的区别与联系。

1.1.1 数据结构的定义

用计算机解决一个具体的问题时,大致需要经过以下几个步骤:

- (1) 分析问题,确定数据模型。
- (2) 设计相应的算法。
- (3) 编写程序,运行并调试程序,直至得到正确的结果。

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,然后用数学语言加以描述。有些问题的数据模型可以用具体的数学方程等来表示,但更多的实际问题是无法用数学方程来表示的,这就需要从数据入手来分析并得到解决问题的方法。

数据是描述客观事物的数、字符以及所有能输入到计算机中并被计算机程序处理的符号的集合。例如,日常生活中使用的各种文字、数字和特定符号都是数据。从计算机的角度看,数据是所有能被输入到计算机中,且能被计算机处理的符号的集合。它是计算机操作的对象总称,也是计算机处理的信息的某种特定的符号表示形式(例如,A班学生数据包含了该班全体学生记录)。

通常以**数据元素**作为数据的基本单位(例如,A班中的每个学生记录都是一个数据元素)。也就是说,数据元素是组成数据的、有一定意义的基本单位,在计算机中通常作为整体处理。有些情况下,数据元素也称为元素、结点和记录等。有时,一个数据元素可以由若干个数据项组成。数据项是具有独立含义的数据最小单位,也称为字段或域(例如,A班中的每个数据元素即学生记录是由学号、姓名、性别和班号等数据项组成的)。

数据对象是性质相同的有限个数据元素的集合,它是数据的一个子集,如大写字母数据对象是集合 $C = \{'A', 'B', 'C', \dots, 'Z'\}$; $1 \sim 100$ 的整数数据对象是集合 $N = \{1, 2, \dots, 100\}$ 。默认情况下,数据结构中的数据都指的是数据对象。

数据结构是指所有数据元素以及数据元素之间的关系,可以看做是相互之间存在着特定关系的数据元素的集合。因此,有时把数据结构看成是带结构的数据元素的集合。数据结构包括如下几个方面:

(1) 数据元素之间的逻辑关系,即数据的逻辑结构,是数据结构在用户面前呈现的形式。

(2) 数据元素及其关系在计算机存储器中的存储方式,即数据的存储结构,也称为数据的物理结构。

(3) 施加在该数据上的操作,即数据的运算。

数据的逻辑结构是从逻辑关系(主要指数据元素的相邻关系)上描述数据的,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做是从具体问题抽象出来的数学模型。

数据的存储结构是逻辑结构用计算机语言的实现或在计算机中的表示(亦称为映像),也就是逻辑结构在计算机中的存储方式,它依赖于计算机语言。一般只在高级语言(如C、C++、C#语言等)的层次上讨论存储结构。

数据的运算是定义在数据的逻辑结构之上的,每种逻辑结构都有一组相应的运算。例如,最常用的运算有检索、插入、删除、更新和排序等。数据的运算最终需在对应的存储结构上用算法实现。

因此,数据结构是一门讨论“描述现实世界实体的数学模型(通常为非数值计算)及其之上的运算在计算机中如何表示和实现”的学科。

1.1.2 数据的逻辑结构

讨论数据结构的目的是为了用计算机求解问题。分析并弄清数据的逻辑结构是求解问题的基础,也是求解问题的第一步。

数据的逻辑结构是用户根据需要建立起来的数据组织形式,它反映数据元素之间的逻辑关系,而不是物理关系,是独立于计算机的。

数据中的数据元素之间可以有不同的逻辑关系,下面通过几个示例加以说明。

【例 1.1】 一个学生的高等数学成绩单见表 1.1。这个表中的数据元素是学生成绩记录,每个数据元素由 3 个数据项(学号、姓名和分数)组成。试讨论其逻辑结构特性。

表 1.1 高等数学成绩单

| 学号 | 姓名 | 分数 | 学号 | 姓名 | 分数 |
|---------|----|----|---------|----|----|
| 2011001 | 王华 | 90 | 2011007 | 许兵 | 76 |
| 2011010 | 刘丽 | 62 | 2011012 | 李萍 | 88 |
| 2011006 | 陈明 | 54 | 2011005 | 李英 | 82 |
| 2011009 | 张强 | 95 | | | |

解: 该表中的每一行称为一个记录,其逻辑结构特性是:只有一个开始记录(即姓名为王华的记录)和一个终端记录(也称为尾记录,即姓名为李英的记录),其余每个记录只有一个前趋记录和一个后继记录。也就是说,记录之间存在一对一的关系。具有这种逻辑特性的逻辑结构称为**线性结构**。

【例 1.2】 某高校组织结构示意图如图 1.1 所示。高校下设若干个学院和若干个处,每个学院下设若干个系,每个处下设若干个科或办公室。试讨论其逻辑结构特性。

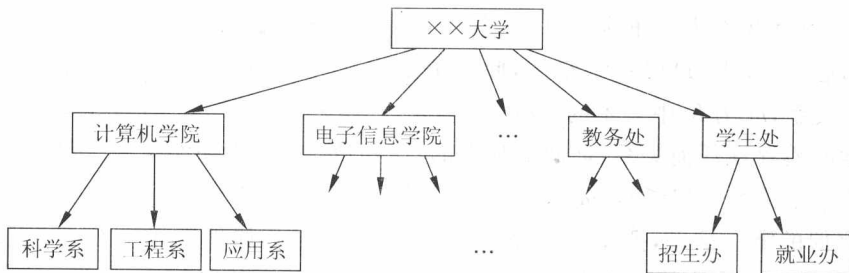


图 1.1 某高校组织结构示意图

解: 该图中的每个长方形框表示一个结点,其逻辑结构特性是:只有一个开始结点(即大学名称结点),有若干个终端结点(如科学系等),每个结点对应零个或多个结点(终端结点对应零个结点)。也就是说,结点之间存在一对多的关系。具有这种逻辑特性的逻辑结构称为**树形结构**。

【例 1.3】 我国部分城市交通线路图如图 1.2 所示。试讨论其逻辑结构特性。

解: 该图中每个城市表示一个结点,其逻辑结构特性是:每个结点与一个或多个结点相连。也就是说,结点之间存在多对多的关系。具有这种逻辑特性的逻辑结构称为**图形结构**。

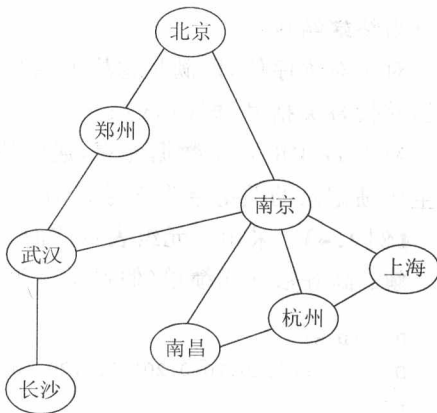


图 1.2 我国部分城市交通线路图

从上面几个示例看出,数据的逻辑结构主要是从数据元素之间的相邻关系来考虑的,数据结构课程中仅讨论这种相邻关系,在实际应用中很容易将其推广到其他关系。

归纳起来,数据的逻辑结构有4种,除了前面介绍的线性结构、树形结构和图形结构外,还有一种集合,集合中的数据元素没有任何相邻关系。

表1.1、图1.1和图1.2分别是高等数学成绩单、某高校组织结构示意图和我国部分城市交通线路图的逻辑结构表示。也就是说,数据逻辑结构可以采用多种方式来表示,假设用学号表示一个成绩记录,高等数学成绩单的逻辑结构也可以用图1.3表示。



图 1.3 高等数学成绩单的另一种逻辑结构表示

为了更通用地描述数据的逻辑结构,通常采用二元组表示数据的逻辑结构。一个二元组如下:

$$B = (D, R)$$

其中, B 是一种数据结构, D 是数据元素的集合,在 D 上数据元素之间可能存在多种关系, R 是所有关系的集合,即:

$$D = \{d_i \mid 1 \leq i \leq n, n \geq 0\}$$

$$R = \{r_j \mid 1 \leq j \leq m, m \geq 0\}$$

其中, d_i 表示集合 D 中的第 i ($1 \leq i \leq n$)个数据元素(或结点), n 为 D 中数据元素的个数,特别地,若 $n=0$,则 D 是一个空集,此时 B 也就无结构可言。 r_j ($1 \leq j \leq m$)表示集合 R 中的第 j 个关系, m 为 R 中关系的个数,特别地,若 $m=0$,则 R 是一个空集,表明集合 D 中的数据元素间不存在任何关系,彼此是独立的,这和数学中集合的概念是一致的。

R 中的某个关系 r_j ($1 \leq j \leq m$)是序偶的集合,对于 r_j 中的任一序偶 $\langle x, y \rangle$ ($x, y \in D$),把 x 叫做序偶的第一结点,把 y 叫做序偶的第二结点,又称序偶的第一结点为第二结点的前趋结点,称第二结点为第一结点的后继结点。如在 $\langle x, y \rangle$ 的序偶中, x 为 y 的前趋结点,而 y 为 x 的后继结点。

若某个结点没有前趋结点,则称该结点为开始结点;若某个结点没有后继结点,则称该结点为终端结点。

对于对称序偶,即满足这样的条件:若 $\langle x, y \rangle \in r$ ($r \in R$),则 $\langle y, x \rangle \in r$ ($x, y \in D$),可用圆括号代替尖括号,即 $(x, y) \in r$ 。

对于 D 中的每个数据元素,通常用一个关键字来唯一标识。例如,高等数学成绩表中学生成绩记录的关键字为学号。前面3个示例均可以采用二元组来表示其逻辑结构。

【例 1.4】 采用二元组表示前面3个例子的逻辑结构。

解: 高等数学成绩单(假设学号为关键字)的二元组表示如下:

$$B_1 = (D, R)$$

$$D = \{2011001, 2011010, 2011006, 2011009, 2011007, 2011012, 2011005\}$$

$$R = \{r_1\}$$

//表示只有一种逻辑关系

$$r_1 = \{\langle 2011001, 2011010 \rangle, \langle 2011010, 2011006 \rangle, \langle 2011006, 2011009 \rangle, \langle 2011009, 2011007 \rangle, \langle 2011007, 2011012 \rangle, \langle 2011012, 2011005 \rangle\}$$