



普通高等教育“十一五”国家级规划教材
国家精品课程主讲教材
教育部高等理工教育教学改革与实践项目研究成果

数据结构 ——用C语言描述

Data Structures in C

耿国华 主编

 高等教育出版社
HIGHER EDUCATION PRESS

普通高等教育“十一五”国家级规划教材
国家精品课程主讲教材
教育部高等理工教育教学改革与实践项目研究成果

数据结构——用 C 语言描述

Shuju Jiegou
——Yong C Yuyan Miaoshu

耿国华 主编
耿国华 张德同 周明全 等编



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书根据教育部计算机科学与技术教学指导委员会制定的“数据结构与算法”课程大纲和《全国硕士研究生入学统一考试计算机专业基础综合考试大纲》的要求而编写,跟踪技术发展需要,教材体系科学,内容简洁实用。

本书共分10章,内容包括基本概念、基本结构(线性表、栈与队列、串、数组与广义表、树、图)和基本技术(查找方法与排序方法)三大部分,每章均附有小结与典型题例,并配有大量的例题、习题与实践题,连贯性强。随书配套光盘教学资源丰富,包括部分扩展内容、课程设计与课程实习指导、多媒体教学课件、算法源码等。

本书既可作为高等学校计算机及相关专业数据结构课程的教材,也可供从事计算机应用开发的工程技术人员参考使用。

图书在版编目(CIP)数据

数据结构:用C语言描述 / 耿国华主编. — 北京:高等教育出版社, 2011.6

ISBN 978 -7 -04 -032485 -3

I. ①数… II. ①耿… III. ①数据结构 - 高等学校 - 教材 ②C语言 - 程序设计 - 高等学校 - 教材
IV. ①TP311.12②TP312

中国版本图书馆CIP数据核字(2011)第217563号

策划编辑 倪文慧 责任编辑 张海波 封面设计 王洋 版式设计 马敬茹
插图绘制 尹莉 责任校对 刘春萍 责任印制 张福涛

| | | | |
|------|-------------------|------|---|
| 出版发行 | 高等教育出版社 | 咨询电话 | 400-810-0598 |
| 社址 | 北京市西城区德外大街4号 | 网 址 | http://www.hep.edu.cn |
| 邮政编码 | 100120 | | http://www.hep.com.cn |
| 印 刷 | 北京市鑫霸印务有限公司 | 网上订购 | http://www.landaco.com |
| 开 本 | 787mm×1092mm 1/16 | | http://www.landaco.com.cn |
| 印 张 | 24.5 | 版 次 | 2011年6月第1版 |
| 字 数 | 550千字 | 印 次 | 2011年6月第1次印刷 |
| 购书热线 | 010-58581118 | 定 价 | 40.00元(含光盘) |

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换
版权所有 侵权必究
物 料 号 32485-00

前 言

IEEE - CS/ACM 的 CS2008 教程已将算法与数据结构课程列为核心课程之首,数据结构愈显出其在信息学科中的重要地位。作为重要的计算机专业核心基础课,数据结构课程一般开设在大学二年级,应对前两年学习的软件技术进行总结和提高,为后续专业课程打牢基础;作为计算机科学与技术人才素质培养框架中的中坚课程,它承上启下,贯通始终,是计算机科学与技术人才素质培养框架中的脊梁,其学习过程体现着创造性思维的信息素质培养过程,对学生的软件开发能力培养至关重要。

本书根据教育部高等学校计算机科学与技术教学指导委员会制定的《高等学校计算机科学与技术专业核心课程教学实施方案》和《全国硕士研究生入学统一考试计算机专业基础综合考试大纲》要求编写。作为普通高等教育“十一五”国家级规划教材,融入了我们建设国家精品课程成果,是在我们 2005 年高等教育出版社出版的《数据结构》教材基础上的更新和提升。跟踪技术发展需要,符合“数据结构与算法”课程大纲要求,教材简洁实用,覆盖了《全国硕士研究生入学统一考试计算机专业基础综合考试大纲》要求内容。

本书以抽象数据类型为中心,采用面向对象的新观点,将教学内容分为基本概念、基本结构、基本技术三个层次,贯穿了计算机科学中的一些重要的问题求解技术,符合认知规律。使用标准 C 作为算法描述的语言,使之与目前大多数院校讲授的第一门编程语言课程衔接,便于将读者的注意力集中在算法的理解上。书中给出了大量的 C 函数表示算法,体现数据抽象与过程抽象的结合,突出了算法表示的实质。

本书内容共分 10 章,包括三大部分,其中第一部分(第 1 章)是数据结构的基本概念部分;第二部分(第 2 ~7 章)是基本的数据结构部分,包括线性结构(线性表、栈与队列、串、数组与广义表)与非线性结构(树、图);第三部分(第 8 ~10 章)是基本技术部分,包括查找方法与排序方法。书中还贯穿了一些重要的程序设计技术,如参数传递技术、动态处理的指针技术、数组技术(抽象规律处理)、递归技术与队列技术;书中给出了许多精彩的典型算法,这是人们在数据处理中智慧的结晶,我们力求将经典算法的思路表现出来,为学习者继续拓展提供线索。

内容特色:

在概念部分,为增强对抽象概念的理解,示例说明数据结构的重要性,突出算法性能评价和问题规模、算法复杂度分析的范例,针对同一个问题给出两个复杂度不同的算法,便于直观地理解算法复杂度的概念和算法复杂度分析的方法。

在结构部分,为便于理解应用抽象数据类型结构,建立完整的结构体系,在学完第 2 章上机实验前可参阅光盘中的“C 语言实现算法模板示例”部分内容,便于引用已有定义类型文件嵌入

相关函数,实现抽象数据类型重用,引导学生掌握算法实现的方法。同时,紧扣考研大纲要求,增补了并查集与等价类划分的内容。

在基本技术部分,突出典型技术的优化处理技术,增加数据结构典型技术的新应用,引入 *B* 树动态索引技术应用示例,便于理解构建现行使用的查找索引数据结构技术;给出了对每类排序方法小结,引导学习者掌握排序思想与改进思路。

每章均附有总结与典型题例,形成梯度,便于总结与提高。与前版教材相比,增大了习题与实习题的比例,对综合性强实习加入部分提示,以突出应用能力培养。

附录给出了近年全国硕士研究生入学统一考试真题及样卷,便于读者模拟练习和考研参考。

资源特色:

鉴于数据结构课程的技术性特点,我们建设充实了与教材配套的多媒体课件和课程教学网站,提供教师、学生及同行之间的课程教学资源与交流平台。

随书所附光盘含有丰富的教学资源,包括面向教学的多媒体课件、面向学习的算法实现和课程设计与实习范例、面向扩展的选学内容三大部分。多媒体课件采用 Flash 动画实现全部算法执行过程,提供的选件控制方式有助于授课内容重新组合,突出数据结构课程难点算法本质的理解,可直接用于课堂讲解。为方便学习和实践,光盘中给出了书中全部所列算法函数实现方式(可执行代码、可装配模板、可引用类型头文件),只需引用相应的类型定义头文件就可直接上机运行。光盘还给出课程实习和设计模板范例,强化实习和设计核心内容。PPT 电子教案可从高等教育出版社网站上免费下载,其他教学资源可访问国家精品课程网站和西北大学精品课程网站。

本书是国家精品课程主讲教材,集作者多年教学实践经验,吸纳了其他院校同行建议,特别得益于国家精品课程建设的支持,随书配套光盘资源逐步丰富完善,形成了面向教学过程与面向学习过程的两大特色。本书由耿国华任主编。本书编写分工如下:第1章、6章、7章、9章及附录由耿国华编写;第5章、8章及实验指导由张德同编写;第3章、10章由周明全编写;第4章由冯宏伟编写,第2章由卢燕宁编写。全书由耿国华统稿,增补工作主要由耿国华、张德同、刘晓宁完成。李康老师负责教学网站建设,多名研究生参加算法调试与动画多媒体课件制作。

本书前版教材已应用于多所学校电子、数学、自动化、通信、计算机专业的教学过程,衷心感谢广大读者对我们前版教材的厚爱与信任,真诚地期待着大家在使用本教材的过程中提出意见和建议,使之能够不断完善,更好地服务于教学。

耿国华

2011年2月

目 录

| | | | |
|------------------------|----|---------------------------------|-----|
| 第 1 章 绪论 | 1 | 基本运算..... | 41 |
| 1.1 数据结构的基础概念 | 1 | 2.3 线性表的链式存储..... | 46 |
| 1.2 数据结构的内容 | 6 | 2.3.1 单链表..... | 46 |
| 1.3 算法 | 9 | 2.3.2 单链表上的基本运算 | 48 |
| 1.4 算法描述..... | 10 | 2.3.3 循环链表..... | 55 |
| 1.5 算法性能评价..... | 11 | 2.3.4 双向链表..... | 57 |
| 1.5.1 算法的时间性能分析 | 12 | 2.3.5 静态链表 | 59 |
| 1.5.2 算法的空间性能分析 | 16 | 2.4 线性表应用——一元多项式的 | |
| 1.5.3 算法性能选择 | 17 | 表示及相加..... | 62 |
| 1.6 数据结构与 C 语言表示 | 17 | 2.5 顺序表与链表的综合比较..... | 66 |
| 1.6.1 数据结构与程序设计的 | | 2.5.1 顺序表和链表的比较 | 66 |
| 关联性..... | 17 | 2.5.2 线性表链式存储方式的 | |
| 1.6.2 结构化程序设计与函数 | | 比较 | 67 |
| 的模块化..... | 18 | 2.6 总结与提高..... | 68 |
| 1.6.3 面向对象与抽象数据类型..... | 20 | 2.6.1 主要知识点 | 68 |
| 1.6.4 算法描述规范与设计风格..... | 25 | 2.6.2 典型题例..... | 69 |
| 1.7 关于学习数据结构..... | 31 | 习题 | 72 |
| 1.8 总结与提高..... | 34 | 实习题 | 74 |
| 习题 | 35 | 第 3 章 限定性线性表——栈与队列 | 76 |
| 实习题 | 36 | 3.1 栈..... | 76 |
| 第 2 章 线性表 | 37 | 3.1.1 栈的定义 | 76 |
| 2.1 线性表的概念及其抽象数据 | | 3.1.2 栈的表示和实现 | 78 |
| 类型定义..... | 37 | 3.1.3 栈的应用举例 | 85 |
| 2.1.1 线性表的逻辑结构 | 37 | 3.1.4 栈与递归的实现 | 89 |
| 2.1.2 线性表的抽象数据 | | 3.2 队列..... | 96 |
| 类型定义..... | 38 | 3.2.1 队列的定义 | 96 |
| 2.2 线性表的顺序存储..... | 39 | 3.2.2 队列的表示和实现 | 97 |
| 2.2.1 线性表的顺序存储结构 | 39 | 3.2.3 队列的应用举例..... | 102 |
| 2.2.2 线性表顺序存储结构上的 | | 3.3 总结与提高 | 105 |

| | | | |
|------------------------------|-----|-------------------------------|-----|
| 3.3.1 主要知识点 | 105 | 6.2 二叉树 | 159 |
| 3.3.2 典型题例 | 106 | 6.2.1 二叉树的定义与基本操作 | 159 |
| 习题 | 108 | 6.2.2 二叉树的性质 | 160 |
| 实习题 | 110 | 6.2.3 二叉树的存储结构 | 162 |
| 第4章 串 | 112 | 6.3 二叉树的遍历与线索化 | 164 |
| 4.1 串的基本概念 | 112 | 6.3.1 二叉树的遍历 | 164 |
| 4.2 串的存储实现 | 114 | 6.3.2 遍历算法应用 | 167 |
| 4.2.1 定长顺序串 | 114 | 6.3.3 基于栈的递归消除 | 172 |
| 4.2.2 堆串 | 120 | 6.3.4 线索二叉树 | 177 |
| 4.2.3 块链串 | 123 | 6.3.5 由遍历序列确定二叉树 | 183 |
| 4.3 串的应用举例:简单的 行编辑器 | 124 | 6.4 树、森林和二叉树的关系 | 184 |
| 4.4 总结与提高 | 125 | 6.4.1 树的存储结构 | 184 |
| 4.4.1 主要知识点 | 125 | 6.4.2 树、森林与二叉树的相互 转换 | 187 |
| 4.4.2 典型题例 | 125 | 6.4.3 树与森林的遍历 | 190 |
| 习题 | 126 | 6.5 哈夫曼树及其应用 | 192 |
| 实习题 | 127 | 6.5.1 哈夫曼树 | 192 |
| 第5章 数组与广义表 | 129 | 6.5.2 哈夫曼编码 | 197 |
| 5.1 数组的定义与运算 | 129 | 6.6 并查集与等价类划分 | 202 |
| 5.2 数组的顺序存储与实现 | 131 | 6.7 总结与提高 | 207 |
| 5.3 特殊矩阵的压缩存储 | 134 | 6.7.1 主要知识点 | 207 |
| 5.3.1 规律分布的特殊矩阵 | 134 | 6.7.2 典型题例 | 208 |
| 5.3.2 稀疏矩阵 | 137 | 习题 | 211 |
| 5.4 广义表 | 146 | 实习题 | 213 |
| 5.4.1 广义表的概念 | 146 | 第7章 图 | 215 |
| 5.4.2 广义表的存储结构 | 147 | 7.1 图的定义与基本术语 | 215 |
| 5.4.3 广义表的操作实现 | 149 | 7.1.1 图的定义 | 215 |
| 5.5 总结与提高 | 151 | 7.1.2 基本术语 | 217 |
| 5.5.1 主要知识点 | 151 | 7.2 图的存储结构 | 220 |
| 5.5.2 典型题例 | 152 | 7.2.1 邻接矩阵表示法 | 220 |
| 习题 | 153 | 7.2.2 邻接表表示法 | 222 |
| 实习题 | 154 | 7.2.3 十字链表 | 225 |
| 第6章 树与二叉树 | 155 | 7.2.4 邻接多重表 | 227 |
| 6.1 树的定义与基本术语 | 155 | 7.3 图的遍历 | 229 |

| | | | |
|-----------------------|-----|------------------------|-----|
| 7.3.1 深度优先搜索 | 229 | 9.2.1 直接插入排序 | 323 |
| 7.3.2 广度优先搜索 | 233 | 9.2.2 折半插入排序 | 325 |
| 7.4 图的应用 | 235 | 9.2.3 希尔排序 | 326 |
| 7.4.1 图的连通性问题 | 235 | 9.2.4 小结 | 329 |
| 7.4.2 有向无环图的应用 | 243 | 9.3 交换类排序法 | 330 |
| 7.4.3 最短路径问题 | 253 | 9.3.1 冒泡排序 | 330 |
| 7.5 总结与提高 | 259 | 9.3.2 快速排序 | 332 |
| 7.5.1 主要知识点 | 259 | 9.3.3 小结 | 335 |
| 7.5.2 典型题例 | 260 | 9.4 选择类排序法 | 336 |
| 习题 | 263 | 9.4.1 简单选择排序 | 336 |
| 实习题 | 266 | 9.4.2 树形选择排序 | 337 |
| 第8章 查找 | 268 | 9.4.3 堆排序 | 339 |
| 8.1 查找的基本概念 | 268 | 9.4.4 小结 | 345 |
| 8.2 基于线性表的查找法 | 269 | 9.5 归并排序 | 346 |
| 8.2.1 顺序查找法 | 269 | 9.6 分配类排序 | 348 |
| 8.2.2 折半查找法 | 270 | 9.6.1 多关键字排序 | 348 |
| 8.2.3 分块查找法 | 273 | 9.6.2 链式基数排序 | 349 |
| 8.3 基于树的查找法 | 275 | 9.6.3 基数排序的顺序表实现 | 353 |
| 8.3.1 二叉排序树 | 275 | 9.7 各种排序方法的综合比较 | 353 |
| 8.3.2 平衡二叉排序树 | 282 | 9.8 总结与提高 | 355 |
| 8.3.3 B树 | 292 | 9.8.1 主要知识点 | 355 |
| 8.4 计算式查找法——哈希法 | 302 | 9.8.2 典型题例 | 355 |
| 8.4.1 哈希函数的构造方法 | 304 | 习题 | 359 |
| 8.4.2 处理冲突的方法 | 306 | 实习题 | 361 |
| 8.4.3 哈希表的查找过程 | 308 | 第10章 外部排序 | 362 |
| 8.4.4 哈希法性能分析 | 309 | 10.1 外排序的基本方法 | 362 |
| 8.5 总结与提高 | 312 | 10.1.1 磁盘排序 | 362 |
| 8.5.1 主要知识点 | 312 | 10.1.2 磁带排序 | 363 |
| 8.5.2 典型题例 | 315 | 10.2 总结与提高 | 363 |
| 习题 | 319 | 习题 | 364 |
| 实习题 | 320 | 附录一 学期样卷 | 365 |
| 第9章 内部排序 | 321 | 学期样卷一 | 365 |
| 9.1 排序的基本概念 | 321 | 学期样卷二 | 367 |
| 9.2 插入类排序 | 322 | 学期样卷三 | 370 |

| | |
|----------------------------------|--------------------------------|
| 附录二 硕士研究生入学考试样题 ····· 373 | 2009 年全国硕士研究生入学统一 |
| 2011 年全国硕士研究生入学统一 | 考试数据结构试题 ····· 376 |
| 考试数据结构试题 ····· 373 | 工程硕士学位研究生模拟试题 ····· 378 |
| 2010 年全国硕士研究生入学统一 | 附录三 光盘目录 ····· 381 |
| 考试数据结构试题 ····· 374 | 参考文献 ····· 382 |

第 1 章 绪 论

陈火旺院士把计算机 60 多年的成就概括为五个“一”：开辟一个新时代——信息时代，形成一个新产业——信息产业，产生一个新学科——计算机科学与技术，开创一种新的科研方法——计算方法，开辟一种新文化——计算机文化，这一概括深刻阐明了计算机对社会发展所产生的广泛而深远的影响。

数据结构和算法是计算机求解问题过程的两大基石。著名的计算机科学家 P. Wegner 指出，“在工业革命中起核心作用的是能量，而在计算机革命中起核心作用的是信息”。计算机科学就是“一种关于信息结构转换的科学”。信息结构(数据结构)是计算机科学研究的基本课题，数据结构又是算法研究的基础。

关于数据结构理论的研究，可以追溯到 1972 年 C. A. R. Hoare 奠基性的论文“数据结构笔记”；而现代计算机所大量采用的各种数据结构，最早的系统论述应归于 D. E. Knuth 的名著《计算机程序设计技巧》。随着计算机科学的飞速发展，数据结构的基础研究也逐渐走向成熟。

在开始讲授数据结构相关内容之前，我们首先需要回答以下问题：

- ① 什么是数据结构？（定义）
- ② 数据结构研究什么？（内容）
- ③ 采用什么方法进行研究？（方法）
- ④ 采用什么工具对算法规则进行描述？（描述）
- ⑤ 如何对算法做性能评价？（评价）
- ⑥ 数据结构和 C 语言的关系是什么？（工具要点、规范要求）
- ⑦ 学习数据结构有什么作用？（重要性）

本章将通过对这些问题的简要介绍，描述数据结构基本内容、主要概念与描述规范，作为本门课程的梗概之序。

1.1 数据结构的基础概念

下面首先介绍数据结构的相关术语。

1. 数据(Data)

数据是描述客观事物的数值、字符以及能输入机器且能被处理的各种符号集合。换句话说，

数据是对客观事物采用计算机能够识别、存储和处理的形式所进行的描述。简而言之,数据就是计算机化的信息。

数据的概念经历了与计算机发展相类似的发展过程。计算机一问世,数据作为程序的处理对象随之产生。早期计算机主要应用于数值计算,数据量小且结构简单,数据仅有进行算术运算与逻辑运算的需求,数据类型只有整型、实型、布尔型,那时程序员把主要精力放在程序设计的技巧上,而并不重视如何在计算机中组织数据。

随着计算机软件、硬件的发展与应用领域的不断扩大,计算机应用领域发生了战略性转移,非数值运算与处理所占的比例越来越高,数据的概念被大大拓展了。它包含数值、字符、声音、图像等一切可以输入到计算机中的符号集合,各种信息通过编码被划归到数据的范畴,大量复杂的非数值数据处理需求使数据的有效组织显得越来越重要。20 世纪 70 年代后,微型计算机的普及以及数据库、人工智能等研究进一步推动了计算机技术的发展,人们越来越重视运用科学工具来探索数据和程序的内部关系以及它们之间的关系,采用新的观点来设计计算机体系,使计算技术发展为一门科学。

数据的概念不再是狭义的,数据已由纯粹的数值概念发展为图像、字符、声音等各种符号。例如,对于 C 语言的编译程序来说,它加工的数据是字符流的源程序(.c),输出的结果是目标程序(.obj);对于 C 语言的链接程序来说,它加工的数据是目标程序(.obj),输出的结果是可执行程序(.exe),如图 1.1 所示。

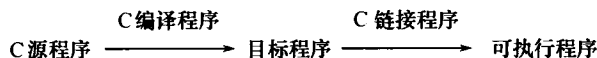


图 1.1 C 编译过程示意图

而对于整个 C 编译过程而言,由于它在操作系统控制下接受操作系统的调度,因此相对操作系统来说它又是数据。

2. 数据元素(Data Element)

数据元素是组成数据的基本单位,是数据集合的个体,在计算机中通常作为一个整体进行考虑和处理。一个数据元素可由一个或多个数据项组成,数据项(Data Item)是有独立含义的最小单位,此时的数据元素通常称为记录(Record)。如表 1.1 所示,学籍表是数据,每一个学生的记录就是一个数据元素。

表 1.1 学 籍 表

| 学 号 | 姓 名 | 性 别 | 籍 贯 | 出生年月 | 住 址 |
|-----|-----|-----|-----|----------|-----|
| 101 | 赵虹玲 | 女 | 河北 | 1983. 11 | 北京 |
| ... | ... | ... | ... | ... | ... |

数据项
 ↓
 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,是数据的一个子集。例如,整数数据对象是集合 $N = \{0, \pm 1, \pm 2, \dots\}$, 字母字符数据对象是集合 $C = \{A, B, \dots, Z\}$, 表 1.1 中所示的学籍表也可看作一个数据对象。由此可看出,不论数据元素集合是无限集(如整数集)、有限集(如字符集),还是由多个数据项组成的复合数据元素(如学籍表),只要性质相同,都属于同一个数据对象。

综上所述,数据概念归纳如下:

数据特点 { 可被计算机接收(与计算机的关联性)
可被加工(能被处理)

数据构成 { 数据元素——组成数据基本单位(是数据集合的个体)
数据对象——性质相同的数据元素的集合(是数据集合的子集)

4. 数据结构 (Data Structure)

数据结构是指相互之间存在一种或多种特定关系的数据元素集合,数据结构应该包括数据元素集合及元素间关系的集合,即数据的组织形式。由此可见,计算机所处理的数据并不是数据的简单汇集,而是具有内在联系的数据集合。例如,图书馆中的图书并不是书的简单堆积,而是带有相应结构关系的书的集合。再如,表结构(如表 1.1 所示的学籍表)、树状结构(如图 1.2 所示的学校组织结构图)、图结构(如图 1.3 所示的交通流量图)。

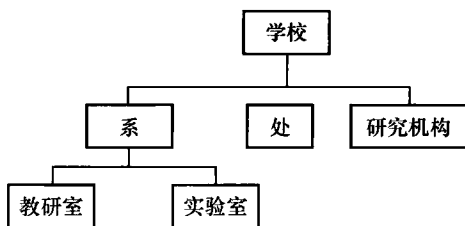


图 1.2 学校组织层次结构图

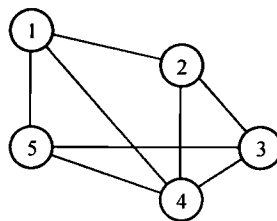


图 1.3 交通流量图

数据结构要研究数据元素之间的相互关系与组织方式,以及对其施加的运算及运算规则,并不关心数据元素的具体内容是什么值。例如,一维数组是向量 $A = (a_1, \dots, a_n)$ 的存储映像,引用时采用下标变量 $A[i]$ 的方式,关注其按序排列、按行存储的特性,并不关心 $A[i]$ 中存放的具体值。同理,二维数组 $A[i, j]$ 是矩阵 $A_{m \times n}$ 的存储映像,我们关心结构关系的特性而不涉及其数组元素本身的内容。

5. 数据类型 (Data Type)

数据类型是一组性质相同的值集合以及定义在这个值集合上的一组操作的总称。数据类型中定义了两个集合,即该类型的取值范围以及该类型中可允许使用的一组运算。例如,高级语言中的数据类型就是已经实现的数据结构的实例。从这个意义上讲,数据类型是高级语言中允许的变量种类,是程序设计语言中已经实现的数据结构(即程序中允许出现的数据形式)。例如在 C 语言中,整型类型的取值范围为 $-32767 \sim +32768$, 可用的运算符集合为加、减、乘、除、取模

(即 +、-、*、/、%)。

从硬件的角度来看,它们的实现涉及字、字节、位、位运算等;从用户观点来看,并不需要了解整数在计算机内如何表示、运算细节如何实现,用户只需要了解整数运算的外部运算特性,就可运用高级语言进行程序设计。引入数据类型的目的,从硬件的角度是将其作为解释计算机内存中信息含义的一种手段,对使用数据类型的用户来说则实现了信息隐蔽,将一切用户不必关心的细节封装在类型中。如两整数求和问题,用户仅关注其数学求和的抽象特性,而不必关心加法运算涉及的内部位运算实现细节。

按“值”的不同特性,一般来说,高级程序设计语言中的数据类型可分为两大类。一类是原子类型,原子类型的值是不可再分的,如 C 语言中提供的整型、实型和字符型及指针;另一类是结构类型,结构类型的值是由若干成分按某种结构组合而成的,因此是可以分解的,并且它的成分可以是原子的,也可以是结构的。例如,数组的值由若干分量组成,每个分量可以是整数,也可以是数组等其他类型。在高级程序设计语言中,数据类型包含系统定义的标准类型和用户自定义类型这两大类,对系统定义的标准类型(如 C 语言中的 int 类型)用户只需按规定的符号形式直接使用,而用户自定义类型必须由用户先定义后使用,用户在系统提供标准类型的基础上根据需要来组合、构造新的类型(如 C 语言中提供的 typedef 定义形式)。

思考题:C 语言中的指针类型属于原子类型还是结构类型?

6. 抽象数据类型

抽象的本质是抽取反映问题的本质点,而忽略非本质的细节,这是从事计算机研究的重要方法。

(1) 数据的抽象

计算机中使用的是二进制数,汇编语言中则可给出各种数据的十进制表示,如 98.65、9.6E3 等,它们是二进制数据的抽象;程序设计人员在编程时可以直接使用,不必考虑实现细节。在高级语言中,则给出更高一级的数据抽象,出现了数据类型,如整型、实型、字符型等。抽象数据类型出现后,可以进一步定义更高级的数据抽象,如各种表、队、栈、树、图、窗口、管理等,这种数据抽象的层次为设计者提供了更有力的手段,使得设计者可以从抽象的概念出发,从整体考虑,然后自顶向下、逐步展开,最后得到所需结果。可以这样看,高级语言中提供整型、实型、字符、记录、文件、指针等多种数据类型,可以利用这些类型构造出像栈、队列、树、图等复杂的抽象数据类型。

(2) 抽象数据类型

抽象数据类型(Abstract Data Type, ADT)定义了一个数据对象、数据对象中各元素间的结构关系以及一组处理数据的操作。从某种意义上讲,抽象数据类型和数据类型实质上是一个概念,只不过 ADT 更广义,不仅限于各种不同的计算机处理器中已定义并实现的数据类型,还包括设计软件系统时用户自己定义的复杂数据类型。

ADT 包括定义和实现两方面,其中定义是独立于实现的。定义仅给出一个 ADT 的逻辑特性,不必考虑如何在计算机中实现。

抽象数据类型的定义取决于客观存在的一组逻辑特性,而与其在计算机内如何表示和实现无关,即不论其内部结构如何变化,只要它的数学特性不变,都不会影响其外部使用,从而为实现软件的部件化和可重用性提供了理论保证,进而提高了软件生产率。

ADT 通常是指由用户定义且用以表示应用问题的数据模型,由基本的数据类型组成,并包括一组相关服务操作。本门课程中将要学习的表、堆栈、队列、串、树、图等结构就是一个个不同的抽象数据类型。以盖楼为例,直接用砖头、水泥、沙子来盖,不仅建造周期长,且建造高度规模受限。如果使用按规范生产成型的水泥预制板,不仅可以高速、安全地建造高楼,水泥预制板使高楼的接缝量大大减少,从而降低了建造高楼的复杂度。由此可见,抽象数据类型是大型软件构造的模块化方法,典型的表、堆栈、队列、串、树、图抽象数据类型,就相当于是设计大型软件的“水泥预制板”,我们用这些已由专门公司设计好的抽象数据类型就可以安全、快速、方便地设计功能复杂的大型软件。

抽象数据类型最重要的特点是数据抽象与信息隐蔽,抽象的本质是抽取反映问题的本质点,忽视非本质的细节,从而使设计的数据结构更具一般性,可以解决一类问题。信息隐蔽就是对用户隐藏数据存储和操作实现的细节,使用者仅需了解操作或界面服务,通过使用界面服务来访问数据,如图 1.4 所示。其中, Request() 用于检查书库,回答读者所借图书是否存在; Retrieve() 用于从书库取书,送书给读者; Return() 用于将书放入书库,将读者所还书放入书架。

抽象数据类型的特征是使用与实现分离,实现封装和信息隐蔽,也就是说,在抽象数据类型设计时,类型的定义与其实现分离。

数据类型的抽象层次越高,含有该抽象数据类型的软件复用程度就越高。ADT 定义该抽象数据类型需要包含哪些信息,并根据功能确定公共界面的服务,使用者可以使用公共界面中的服务对该抽象数据类型进行操作。从使用者的角度看,只要了解该抽象数据类型的规格说明,就可以利用其公用界面中的服务来使用这个类型,不必关心其物理实现,从而集中考虑如何解决实际问题。

ADT 物理实现作为私有部分封装在其实现模块内,使用者不能看到,也不能直接操作该类型所存储的数据,只有通过界面中的服务来访问这些数据。从实现者的角度来看,把抽象数据类型的物理实现封装起来,有利于编码、测试,也有利于修改。当需要改进数据结构时,只要界面服务的使用方式不变,只要改变抽象数据类型的物理实现,而无须修改所有使用该抽象数据类型的程序,这样就会提高系统的稳定性。

下面给出问题求解过程的两个示例,从中体会数据结构对解决问题的重要作用。

例 1.1 电话号码查询问题。

【问题描述】

编写程序实现对某市或某单位个人电话信息的查询功能。具体要求如下:

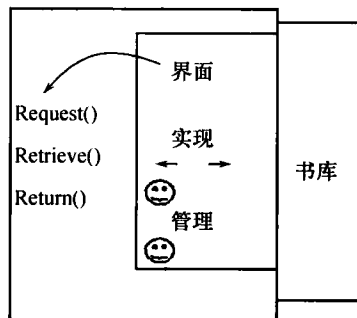


图 1.4 信息隐蔽示意图

1. 按姓名进行查询。
2. 若该人装有电话,则显示其电话号码;若尚未安装,则显示提示信息。

【问题抽象】 从实际问题出发,可将市内(单位内)已装电话人员信息抽象为如表 1.2 所示结构。

表 1.2 电话簿结构

| 姓 名 | 电 话 号 码 |
|-----|---------|
| 张三 | 8830259 |
| 王四 | 8830251 |

【求解方法】 可有顺序表法及索引结构法两种求解方法。

顺序表法:可用数组存储整个表,每个人的信息(即一个人的姓名和电话号码)就是数组中的一个元素。从表的第一项开始依次查找姓名,直至找到或是确定无此姓名为止。这种依次查找方法对于一个规模不大的单位是可行的,但对拥有上百万人电话的城市来说效率太低,应当构建新的数据结构支持快速查找。

索引结构法:采用类似于汉语字典的组织形式,按姓氏拼音或姓氏笔画排列构建表,并构造其姓氏或笔画索引表。查找时只需先在索引表中查找姓氏,然后根据索引表指示在登记表中核查姓名,从而提高了查找效率,如图 1.5 所示。

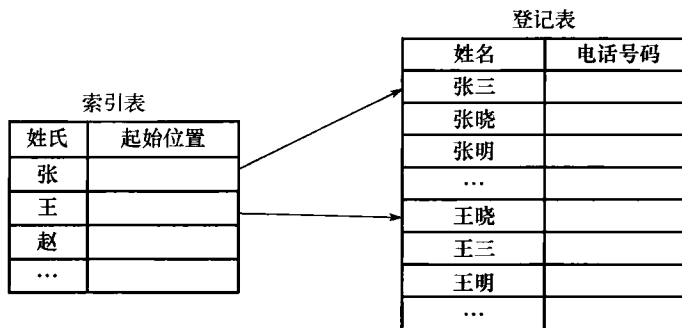


图 1.5 索引电话簿

1.2 数据结构的内容

在 1.1 节中已给出了数据结构初步概念,数据结构是指相互之间存在一种或多种特定关系的数据元素集合。这个描述是一种非常简单的解释。数据元素间的相互关系具体应包括三个方

面:数据的逻辑结构、数据的物理结构和数据的运算集合。

1. 逻辑结构

数据的逻辑结构是指数据元素之间逻辑关系描述。

数据结构的形式定义为:数据结构是一个二元组

$$\text{Data_Structure} = (D, R)$$

其中, D 是数据元素的有限集, R 是 D 上关系的有限集。

例如,

$$DS_2 = (D_2, R_2)$$

$$D_2 = \{ a, b, c, d, e, f \}$$

$$R_2 = \{ T \}$$

$$T = \{ \langle a, b \rangle, \langle a, c \rangle, \langle a, d \rangle, \langle c, e \rangle, \langle c, f \rangle \}$$

则其逻辑结构图为一棵树。

根据数据元素之间关系的不同特性,通常有下列四类基本结构(如图 1.6 所示)。

- ① **集合结构**:结构中的数据元素之间除了同属于一个集合的关系外,无任何其他关系。
- ② **线性结构**:结构中的数据元素之间存在着一对一的线性关系。
- ③ **树状结构**:结构中的数据元素之间存在着一对多的层次关系。
- ④ **图状结构或网状结构**:结构中的数据元素之间存在着多对多的任意关系。

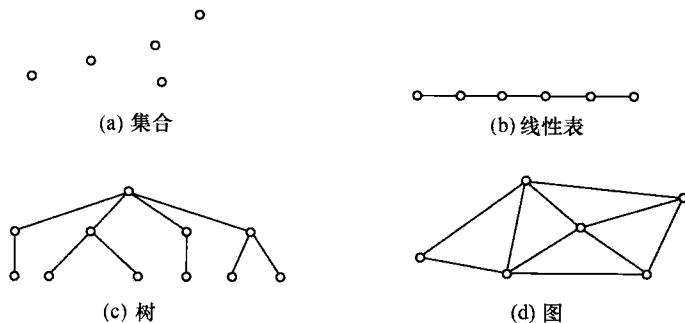


图 1.6 四类基本数据结构示意图

由于集合中关系只有属于或不属于这种简单的属于关系,可以用其他的结构代替它,故数据的四类基本逻辑结构可概括如下:

逻辑结构 $\begin{cases} \text{线性结构——线性表、栈、队、字符串、数组、广义表} \\ \text{非线性结构——树、图} \end{cases}$

2. 存储结构

存储结构(又称物理结构)是逻辑结构在计算机中的存储映像,包括数据元素映像和关系映像,它是逻辑结构在计算机中的实现,它包括数据元素的表示和关系的表示。

形式化描述:对于一种数据结构 (D, R) ,要将 D 存入计算机中,建立一种从 D 的数据元素到存储空间 M 单元的映像 S ,即 $D \rightarrow M$,也就是说对于每一个 $d, d \in D$,都有唯一的 $z \in M$ 使 $S(D) = Z$,同时这个映像必须明显或隐含地体现关系 R 。

逻辑结构与存储结构的关系为:存储结构是逻辑关系的映像与元素本身的映像。逻辑结构是数据结构的抽象,存储结构是数据结构的实现,两者综合起来建立了数据元素之间的结构关系。

数据元素之间的关系在计算机中有两种不同的表示方法:

- 顺序映像(顺序存储结构)
- 非顺序映像(非顺序存储结构)

关系映像在计算机中可用顺序存储结构或非顺序存储结构这两种不同表示方式来存放。逻辑结构在计算机存储器中实现时,可采用不同的存储器来存储,不论是在内存或外存中存储,都要以反映逻辑关系为原则。

3. 运算集合

讨论数据结构的目的是为了在计算机中实现操作,因此在结构上的运算集合是很重要的部分。数据结构就是研究一类数据的表示及其相关的运算操作。

通过如表 1.3 所示工资表实例对数据结构的内容作一概括。

表 1.3 工 资 表

| 编 号 | 姓 名 | 性 别 | 基本工资 | 工 龄 工 资 | 应扣工资 | 实发工资 |
|---------|-----|-----|-----------|---------|---------|-----------|
| 100001 | 张爱芬 | 女 | 3 545. 67 | 245. 45 | 50. 00 | 3 741. 12 |
| 100002 | 李林 | 男 | 3 645. 90 | 285. 60 | 55. 00 | 3 876. 50 |
| 100003 | 刘晓峰 | 男 | 3 645. 00 | 230. 00 | 65. 00 | 3 810. 00 |
| 100004 | 赵俊 | 女 | 3 560. 90 | 325. 90 | 70. 00 | 3 816. 80 |
| 100005 | 孙涛 | 男 | 3 650. 60 | 290. 80 | 60. 00 | 3 881. 40 |
| ... | ... | ... | ... | ... | ... | ... |
| 1000121 | 张兴强 | 男 | 3 725. 98 | 465. 53 | 150. 00 | 4 041. 51 |

在表 1.3 所示工资表中,采用了线性表的逻辑结构,因为结点与结点之间是一种简单的线性关系;由于工资表可能包括几千名职工信息,可采用顺序结构存放,也可采用非顺序结构存放。怎么存放就是具体存储结构的问题了。对于工资表,当职工调离时要删除相应的数据元素,调进时要增加数据元素,调整工资时要修改数据元素。这里的增、删、改就是数据的操作集合。

综上所述,数据结构的内容可归纳为三个部分:逻辑结构、存储结构和运算集合。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们存放在计算机的存储器中,并在这些数据上定义一个运算的集合,这些是数据结构课程的基本内容。

数据结构主要研究怎样合理地组织数据、建立合适的结构、提高执行程序所用的时空效率。