



高职高专“十二五”规划教材

COMPUTER

数据结构

主编 张珊靓 赵浩婕



吉林大学出版社

内容简介

《数据结构》是计算机专业的一门重要课程，也是计算机专业的一门基础课程。本书是根据教育部颁布的《计算机专业人才培养方案》和《计算机专业教学大纲》的要求，结合编者多年的教学经验和研究成果编写而成的。本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机工作的工程技术人员参考。

高职高专“十二五”规划教材

数据结构

主 编 张珊靓 赵浩婕

副主编 柏 强 朱小云 李 鹏

吉林大学出版社

吉林出版集团
吉林出版集团有限责任公司
地址：长春市人民大街4646号
邮编：130021
电话：0431-85164722
E-mail: jlu@nortel.com.cn

内容简介

数据结构主要研究数据间的联系（数据的逻辑结构）、数据在计算机中的存储方法（数据的物理结构）以及处理不同结构数据的算法。

本书内容共分九章，第一章介绍数据结构的基本概念与算法的表示和分析方法，是在先导课程“高级语言程序设计”基础上的进一步引申和总结；第二章到第七章以线性结构、层次结构、网状结构为主线，由简到繁地介绍了顺序表、链表以及栈和队、树和二叉树、图等几种基本数据结构及有关算法；第八章和第九章详细介绍了在实际应用中广泛使用的查找和排序的基本算法。

全书以知识单元为结构，各知识单元内容相互独立，知识面广，内容丰富。本教材的主要对象是非计算机应用专业的大学本科、专科学生，亦可供从事计算机应用工作的管理人员和技术人员学习参考。

图书在版编目（CIP）数据

数据结构 / 张珊靓, 赵浩婕主编. —长春: 吉林大学出版社, 2009.12

（高职高专“十二五”规划教材）

ISBN 978-7-5601-5187-8

I. ①数… II. ①张… ②赵… III. ①数据结构—高等学校: 技术学校—教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字（2009）第 240311 号

书 名: 高职高专“十二五”规划教材
数据结构

作 者: 张珊靓 赵浩婕 主编

责任编辑、责任校对: 邵宇彤

吉林大学出版社出版、发行

开本: 787×1092 毫米 1/16

印张: 14.75 字数: 342 千

ISBN 978-7-5601-5187-8

封面设计: 超视觉工作室

北京市彩虹印刷有限责任公司 印刷

2010 年 2 月 第 1 版

2012 年 6 月 第 2 次印刷

定价: 28.00 元

版权所有 翻印必究

社址: 长春市明德路 501 号 邮编: 130021

发行部电话: 0431-89580026/28/29

网址: <http://www.jlup.com.cn>

E-mail: jlup@mail.jlu.edu.cn

出版说明

作为高等教育的重要组成部分,高等职业教育是以培养具有一定理论知识和较强实践能力,面向生产、面向服务和管理第一线职业岗位的实用型、技能型专门人才为目的的职业技术教育,是职业技术教育的高等阶段。目前,高等职业教育教学改革已经从专业建设、课程建设延伸到了教材建设层面。根据国家教育部关于要求发展高等职业技术教育,培养职业技术人才的大纲要求,我们组织编写了这套《高职高专“十二五”规划教材》。本系列教材坚持以就业为导向,以能力为本位,以服务学生职业生涯发展为目的的指导思想,以与专业建设、课程建设、人才培养模式同步配套作为编写原则。

从专业建设角度,相对于普通高等教育的“学科性专业”,高等职业教育属于“技术性专业”。技术性专业的知识往往由与高新技术工作相关联的那些学科中的有关知识所构成,这种知识必须具有职业技术岗位的有效性、综合性和发展性。本套教材不但追求学科上的完整性、系统性和逻辑性,而且突出知识的实用性、综合性,把职业岗位所需要的知识和实践能力的培养融会于教材之中。

从课程建设角度,现有的高等职业教育教材从教育内容上需要改变“重理论轻实践”、“重原理轻案例”,教学方法上则需要改变“重传授轻参与”、“重课堂轻现场”,考核评价上则需改变“重知识的记忆轻能力的掌握”、“重终结性的考试轻形成性考核”的倾向。针对这些情况,本套教材力求在整体教材内容体系以及具体教学方法指导、练习与思考等栏目中融入足够的实训内容,加强实践性教学环节,注重案例教学,注重能力的培养,使职业能力的培养贯穿于教学的全过程。同时,使公共基础类教材突出职业化,强调通用能力、关键能力的培养,以推动学生综合素质的提高。

从人才培养模式角度,高等职业教育人才的培养模式的主要形式是产学结合、工学交替。因此,本教材为了满足有学就有练、学完就能练、边学边练的实际要求,纳入新技术引用、生产案例介绍等来满足师生教学需要。同时,为了适应学生将来因为岗位或职业的变动而需要不断学习的情况,教材的编写注重采用新知识、新工艺、新方法、新标准,同时注重对学生创造能力和自我学习能力的培养,力争实现学生毕业与就业上岗的零距离。

为了更好地落实指导思想和编写原则,本套教材的编写者既有一定的教学经验、懂得教学规律,又有较强的实践技能。同时,我们还聘请生产一线的技术专家来审稿,保证教材的实用性、先进性、技术性。总之,该套教材是所有参与编写者辛勤劳作和不懈努力的成果,希望本套教材能为职业教育的提高和发展作出贡献。

这就是我们编写这套教材的初衷。

前 言

数据结构对计算机学科起到承前启后的作用，它是计算机专业重要的专业基础课程之一，也是计算机及相关专业考研和水平等级考试的必考科目，而且正逐渐发展成为众多理工专业的热门选修课。

计算机编程中加工处理的对象是数据，而数据具有一定的组织结构，所以学习编写计算机程序仅仅了解计算机语言是不够的，还必须掌握数据组织、存储和运算的一般方法，这便是数据结构课程中所学习和研究的内容，也是我们编写计算机程序的重要基础。

由于数据结构的原理和算法较抽象，数据结构课程知识丰富，内容抽象，学习量大，隐藏在各部分内容中的方法和技术多，而该课程一般在本科低年级开设，对于具有一些计算机程序设计知识的初学者，理解和掌握其中的原理就困难了。编者根据多年的教学经验，在分析国内多种同类教材的基础上，博采众长，编写了这本书，奉献给广大读者。

本书具有以下几个特点：

- (1) 用类 C 语言描述数据结构和算法。
- (2) 通俗易懂，由浅入深。
- (3) 注重理论与实践相结合。

本书由张珊靓、赵浩婕任主编，柏强、朱小云、李鹏任副主编。其中第一、二章由朱小云编写；第三、四章由赵浩婕编写；第六章由柏强（三亚理工职业学院）编写；第五、七章由张珊靓编写；第八、九章由李鹏编写。

由于作者水平有限，书中不妥与疏漏之处难免，敬请广大读者批评指正。

编 者

2009 年 3 月

目 录

第一章 数据结构概论	1
第一节 数据结构研究的主要内容	1
第二节 数据结构的重要性	2
第三节 基本概念和术语	6
第四节 算法和算法分析	10
习 题	16
第二章 线性表	18
第一节 线性表及其运算	18
第二节 线性表的顺序存储	19
第三节 线性表的链式存储	23
第四节 线性表的应用举例	36
习 题	41
第三章 栈和队列	45
第一节 栈	45
第二节 栈的应用——栈和递归	49
第三节 队 列	59
第四节 队列的应用举例	66
习 题	73
第四章 串	75
第一节 串的定义及其运算	75
第二节 串的存储	76
第三节 串的模式匹配	84
习 题	89
第五章 数组和广义表	92
第一节 数 组	92
第二节 广义表	95
习 题	102

第六章 树	104
第一节 树的定义和基本术语	104
第二节 二叉树	106
第三节 遍历二叉树	111
第四节 线索二叉树	123
第五节 霍夫曼树	127
第六节 树和森林	132
习 题	139
第七章 图	140
第一节 图的定义和术语	140
第二节 图的存储结构	142
第四节 图的遍历	150
第四节 生成树和最小生成树	154
第五节 拓扑排序与关键路径	160
第六节 最短路径	168
习 题	172
第八章 查找	174
第一节 查找的基本定义和术语	174
第二节 静态查找	175
第三节 动态查找	179
第四节 哈希表	189
习 题	195
第九章 排序	196
第一节 排序的基本概念	196
第二节 插入排序	196
第三节 交换排序	202
第四节 选择排序	206
第五节 归并排序	212
第六节 基数排序	213
第七节 各种内排序方法的比较	217
第八节 外部排序	217
习 题	224
参考文献	225

第一章 数据结构概论

自从 1946 年第一台计算机问世以来,计算机技术的发展日新月异,其应用已不再局限于科学计算,而是更多地用于控制、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。数据结构就是研究数据组织、存储和运算的一般方法的学科。本章讨论数据结构的基本概念、算法及相关题解。

计算机科学是一门研究数据表示和数据处理的科学。数据是计算机化的信息,它是计算机可以直接处理的最基本和最重要的对象。无论是进行科学计算或数据处理、过程控制,还是对文件的存储和检索及数据库技术等,都是对数据进行加工处理的过程。因此,要设计出一个结构好、效率高的程序,必须研究数据的特性及数据间的相互关系及其对应的存储表示,并利用这些特性和关系设计出相应的算法和程序。

第一节 数据结构研究的主要内容

数据结构与数学、计算机硬件和软件有十分密切的关系。数据结构是介于数学、计算机硬件和计算机软件之间的一门计算机科学与技术专业的核心课程,是高级语言程序设计、编译原理、操作系统、数据库技术、人工智能等课程的基础。同时,数据结构技术也广泛应用于信息科学、系统工程、应用数学以及各种工程技术领域。

数据结构课程集中讨论软件开发过程中的设计阶段,同时涉及编码和分析阶段的若干基本问题。此外,为了构造出好的数据结构及其实现,还需考虑数据结构及其实现的评价与选择。因此,数据结构的内容包括三个层次的五个“要素”,如表 1-1 所示。

表 1-1 数据结构课程内容体系

层次 \ 方面	数据表示	数据处理
抽象	逻辑结构	基本运算
实现	存储结构	算法
评价	不同数据结构的比较及算法分析	

数据结构的核​​心技术是分解与抽象。通过分解可以划分出数据的三个层次;再通过抽象,舍弃数据元素的具体内容,就得到逻辑结构。类似地,通过分解将处理要求划分成各种功能,再通过抽象舍弃实现细节,就得到运算的定义。上述两个方面的结合使我们将问题变换为数据结构。这是一个从具体(即具体问题)到抽象(即数据结构)的过程。然后,通过增加对实现细节的考虑进一步得到存储结构和实现运算,从而完成设计任务。这是一个从抽

象(即数据结构)到具体(即具体实现)的过程。熟练地掌握这两个过程是数据结构课程在专业技能培养方面的基本目标。

数据结构作为一门独立的课程在国外是从 1968 年才开始的,但在此之前其有关内容已散见于编译原理及操作系统之中。20 世纪 60 年代中期,美国的一些大学开始设立有关课程,但当时的课程名称并不叫数据结构。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。从 20 世纪 70 年代中期到 80 年代,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,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,越来越被人们所重视。

第二节 数据结构的重要性

在计算机发展的初期,人们使用计算机的目的主要是处理数值计算问题。当我们使用计算机来解决一个具体问题时,一般需要经过下列几个步骤:首先要从该具体问题抽象出一个适当的数学模型,然后设计或选择一个解此数学模型的算法,最后编出程序进行调试、测试,直至得到最终的解答。例如,求解梁架结构中应力的数学模型的线性方程组,该方程组可以使用迭代算法来求解。

由于当时所涉及的运算对象是简单的整型、实型或布尔类型数据,所以程序设计者的主要精力是集中于程序设计的技巧上,而无须重视数据结构。随着计算机应用领域的扩大和软件、硬件的发展,非数值计算问题越来越显得重要。据统计,当今处理非数值计算性问题占用了 90% 以上的机器时间。这类问题涉及的数据结构更为复杂,数据元素之间的相互关系一般无法用数学方程式加以描述。因此,解决这类问题的关键不再是数学分析和计算方法,而是要设计出合适的数据结构。

数据结构与其他课程之间的关系如图 1-1 所示。

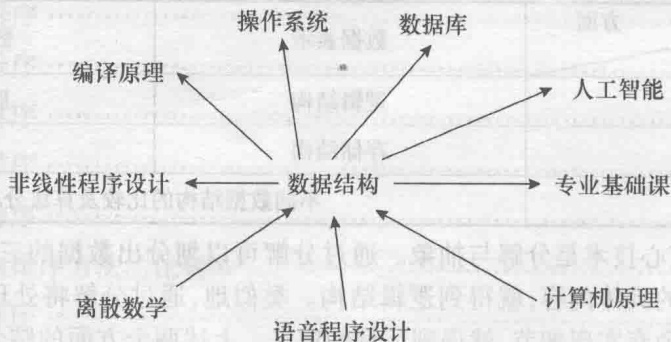


图 1-1 数据结构与其他课程之间的关系

明确提出数据结构概念不过 30 多年,“数据结构”作为一门独立课程在国外于 1968 年开始设立,我国从 20 世纪 80 年代初才开始正式开设“数据结构”课程。“数据结构”课程较系统地介绍了软件设计中常用数据结构以及相应的存储结构和算法,系统介绍了常用的查找和排序技术,并对各种结构与技术进行分析和比较,内容非常丰富。数据结构涉及多方面的知识,如计算机硬件范围的存储装置和存取方法,在软件范围中的文件系统、数据的动态管理、信息检索,数学范围的集合、逻辑的知识,还有一些综合性的知识,如数据类型、程序设计方法、数据表示、数据运算、数据存取等,它是计算机专业的一门重要的专业技术基础课程。“数据结构”的内容将为操作系统、数据库原理、编译原理等后续课程的学习打下良好的基础。“数据结构”课程不仅讲授数据信息在计算机中的组织和表示方法,同时也训练学生高效地解决复杂问题程序设计的能力,因此“数据结构”是数学、计算机硬件、计算机软件三者之间的一门核心课程,是计算机专业提高软件设计水平的一门关键性课程。

下面所列举的就是属于这一类的具体问题。

例 1-1

学生信息检索系统。当我们需要查找某个学生的有关情况的时候,或者想查询某个专业或年级的学生的有关情况的时候,只要建立了相关的数据结构,按照某种算法编写了相关程序,就可以实现计算机自动检索。由此,可以在学生信息检索系统中建立一张按学号顺序排列的学生信息表和分别按姓名、专业、年级顺序排列的索引表,如表 1-2、表 1-3、表 1-4、表 1-5 所示。由这四张表构成的文件便是学生信息检索的数学模型,计算机的主要操作就是按照某个特定要求(如给定姓名)对学生信息文件进行查询。

表 1-2

学生信息表

序号	学号	姓名	性别	专业	年级
1	200603001	张承国	男	计算机科学与技术	06 级
2	200604006	刘 芳	女	信息与计算科学	06 级
3	200702019	崔 丽	女	数学与应用数学	07 级
4	200704026	何淑芬	女	信息与计算科学	07 级
5	200703012	赵志国	男	计算机科学与技术	07 级
6	200803001	刘 芳	女	计算机科学与技术	08 级
7	200802006	魏 龙	男	数学与应用数学	08 级
8	200903003	周 桐	男	计算机科学与技术	09 级
9	200904022	李国勤	男	信息与计算科学	09 级

表 1-3

姓名索引表

崔 丽	3
何淑芬	4
李国勤	9
刘 芳	2,6
魏 龙	7
张承国	1
周 桐	8
赵志国	5

表 1-4

专业索引表

计算机科学与技术	1,5,6,8
信息与计算科学	2,4,9
数学与应用数学	3,7

表 1-5

年级索引表

2006 级	1,2
2007 级	3,4,5
2008 级	6,7
2009 级	8,9

诸如此类的还有电话自动查号系统、考试查分系统、仓库库存管理系统等。在这类文档管理的数学模型中,计算机处理的对象之间通常存在着一种简单的线性关系,这类数学模型可称为线性的数据结构。

例 1-2

八皇后问题。在八皇后问题中,处理过程不是根据某种确定的计算法则,而是利用试探和回溯的探索技术求解。为了求得合理布局,在计算机中要存储布局的当前状态。从最初的布局状态开始,一步步地进行试探,每试探一步形成一个新的状态,整个试探过程形成了一棵隐含的状态树,如图 1-2 所示(为了描述方便,将八皇后问题简化为四皇后问题)。回溯法求解过程实质上就是一个遍历状态树的过程。在这个问题中所出现的树也是一种数据结构,它可以应用在许多非数值计算的问题中。

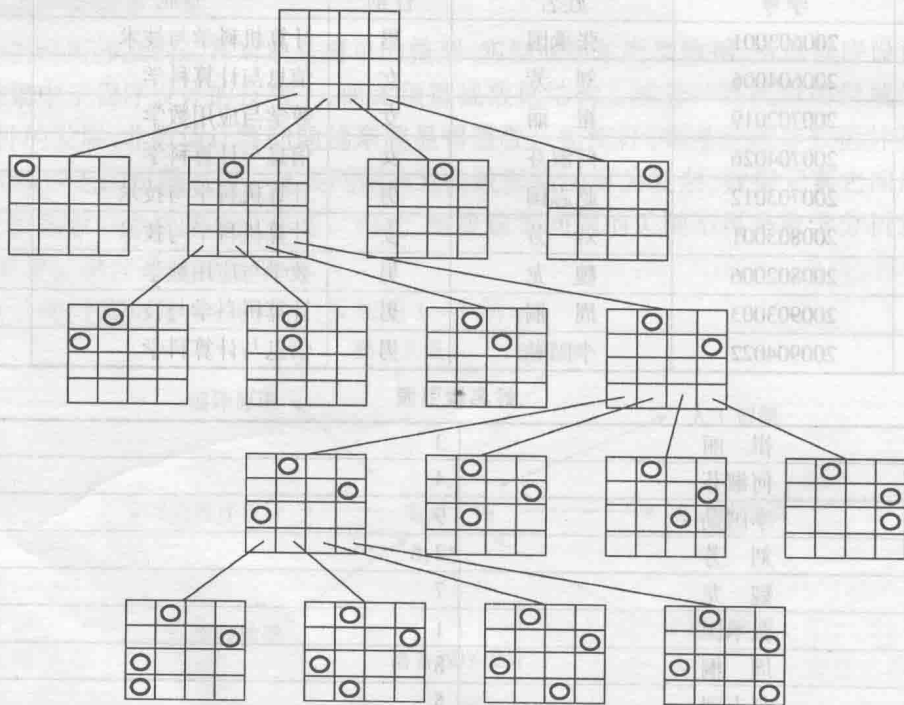


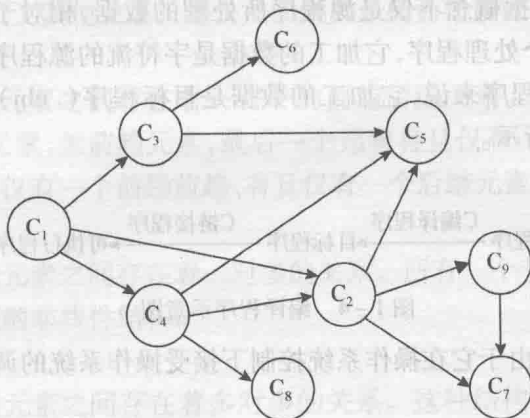
图 1-2 四皇后问题中隐含的状态树

例 1-3

教学计划编排问题。一个教学计划包含许多课程,在教学计划包含的许多课程之间,有些必须按规定的先后次序进行,有些则没有次序要求。即有些课程之间有先修和后续的关系,有些课程可以任意安排次序。这种各个课程之间的次序关系可用一个称做图的数据结构来表示,如图 1-3 所示。有向图中的每个顶点表示一门课程,如果从顶点 C_i 到 C_j 之间存在有向边 $\langle C_i, C_j \rangle$,则表示课程 i 必须先于课程 j 进行。

课程编号	课程名称	先修课程
C_1	计算机导论	无
C_2	数据结构	C_1, C_4
C_3	汇编语言	C_1
C_4	C 程序设计语言	C_1
C_5	计算机图形学	C_2, C_3, C_4
C_6	接口技术	C_3
C_7	数据库原理	C_2, C_9
C_8	编译原理	C_4
C_9	操作系统	C_2

(a) 计算机专业的课程设置



(b) 表示课程之间优先关系的有向图

图 1-3 教学计划编排问题的数据结构

由以上三个例子可见,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、图之类的数据结构。因此,可以说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的学科。

数据结构的发展趋势包括两个方面:一是面向专门领域中特殊问题的数据结构的研究和发展,如图形数据结构、知识数据结构、空间数据结构;另一方面,从抽象数据类型的角度出发,用面向对象的观点来讨论数据结构,已成为新的发展趋势。

学习数据结构的目的是了解计算机处理对象的特性,将实际问题中所涉及的处理对象在计算机中表示出来并对它们进行处理。与此同时,通过算法训练来提高学生的思维能力,通过程序设计的技能训练来促进学生的综合应用能力和专业素质的提高。

第三节 基本概念和术语

数据结构作为一门学科,主要研究数据的各种逻辑结构和存储结构,以及对数据的各种操作。因此,主要有三个方面的内容:数据的逻辑结构、数据的物理存储结构、对数据的操作(即算法)。通常,算法的设计取决于数据的逻辑结构,算法的实现取决于数据的物理存储结构。数据结构的研究不仅涉及计算机硬件的研究,比如存储装置和存取方法,而且是解决编译原理、操作系统、数据库系统的数据元素在存储器中的分配问题的重要基础。在系统地学习数据结构知识之前,先对一些基本概念和术语赋予确切的含义。

一、数据

数据(Data)是信息的载体,它能够被计算机识别、存储和加工处理。它是计算机程序加工的原料,应用程序处理各种各样的数据。计算机科学中,所谓数据就是计算机加工处理的对象,它可以是数值数据,也可以是非数值数据。数值数据是一些整数、实数或复数,主要用于工程计算、科学计算和商务处理等;非数值数据包括字符、文字、图形、图像、语音等。

例如对 C 源程序,数据概念不仅是源程序所处理的数据,相对于编译程序来说,C 编译程序相对于源程序是一个处理程序,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 1-4 所示。

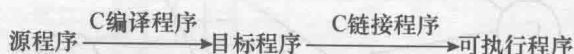


图 1-4 编译程序示意图

而对于 C 编译程序,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。

二、数据元素

数据元素(Data Element)是数据的基本单位。在不同的条件下,数据元素又可称为元素、结点、顶点、记录等。例如,学生信息检索系统中学生信息表的一个记录、八皇后问题中状态树的一个状态、教学计划编排问题中的一个顶点等,都被称为一个数据元素。

有时,一个数据元素可由若干个数据项(Data Item)组成。例如,学籍管理系统中学生信息表的每一个数据元素就是一个学生记录,它包括学生的学号、姓名、性别、籍贯、出生年月、成绩等数据项。这些数据项可以分为两种:一种叫做初等项,如学生的性别、籍贯等,这些数据项是在数据处理时不能再分割的最小单位;另一种叫做组合项,如学生的成绩,它可以再划分为数学、物理、化学等更小的项。通常,在解决实际问题时是把每个学生记录当作一个基本单位进行访问和处理的。

三、数据对象

数据对象(Data Object)或数据元素类(Data Element Class)是具有相同性质的数据元素的集合。在某个具体问题中,数据元素都具有相同的性质(元素值不一定相等),属于同一数据对象(数据元素类),数据元素是数据元素类的一个实例。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,都是同一个数据对象。例如,在交通咨询系统的交通网中,所有的顶点是一个数据元素类,顶点A和顶点B各自代表一个城市,是该数据元素类中的两个实例,其数据元素的值分别为A和B。

四、数据结构

数据结构(Data Structure)是指互相之间存在着一种或多种关系的数据元素的集合。在任何问题中,数据元素之间都不会是孤立的,在它们之间都存在着这样或那样的关系,这种数据元素之间的关系称为结构。根据数据元素间关系的不同特性,通常有下列四类基本的结构。

1. 集合结构

在集合结构中,数据元素间的关系是“属于同一个集合”,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。集合是元素关系极为松散的一种结构。

2. 线性结构

在线性结构中,数据元素之间存在着一对一的前趋后继关系。在此种结构下:第一个元素有且仅有一个后继元素,无前趋元素;最后一个元素有且仅有一个前趋元素,无后继元素;其余每个元素均有且仅有一个前趋前趋、有且仅有一个后继元素。

3. 树型结构

在树型结构中,数据元素之间存在着一对多的关系。所有一个节点最多有一个前趋,能有多多个后继,是一种典型的非线性结构。

4. 图形结构

在图形结构中,数据元素之间存在着多对多的关系。这种结构的特征是所有一个元素能有多多个前趋,也能有多多个后继,是一种多对多的前趋后继关系。图形结构也称网状结构。

图1-5为表示上述四类基本结构的示意图。由于集合是数据元素之间关系极为松散的一种结构,因此也可用其他结构来表示它。

从上面所介绍的数据结构的概念可以知道,一个数据结构有两个要素:一个是数据元素的集合,另一个是关系的集合。在形式上,数据结构通常可以采用一个二元组来表示。

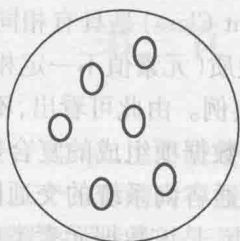
数据结构的形式定义为:数据结构是一个二元组

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

其中,K是数据元素的有限集,R是K上关系的有限集,而每个关系都是从K到K的关系。

设 r 是一个K到K的关系, $r \in R$,若 $k, k' \in K$,且 $k' < k, k' > r$,则称 k' 是 k 的后继, k 是 k' 的前驱,这时 k 和 k' 是相邻的结点(相对 r 而言);如果不存在一个 k' 使 $k' < k, k' > r$,则称

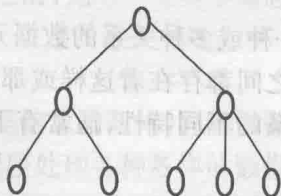
k 为 r 的终端结点;如果不存在一个 k' 使 $k < k', k > r$, 则称 k 为 r 的开始结点;如果 k 既不是终端结点也不是开始结点, 则称 k 是内部结点。



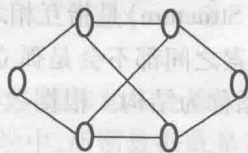
(a)集合结构



(b)线性结构



(c)树型结构



(d)图型结构

图 1-5 四类基本结构的示意图

五、数据的逻辑结构和数据的物理结构

数据结构包括数据的逻辑结构和数据的物理结构。数据的逻辑结构可以看做从具体问题抽象出来的数学模型,它与数据的存储无关。我们研究数据结构的目的是在计算机中实现对它的操作,为此还需要研究如何在计算机中表示一个数据结构。数据结构在计算机中的标识(又称映像)称为数据的物理结构,或称存储结构。它所研究的是数据结构在计算机中的实现方法,包括数据结构中元素的表示及元素间关系的表示。

六、数据的存储结构

数据的存储结构是数据的逻辑结构在计算机存储器中的实现,逻辑结构是从逻辑关系上观察数据,它与数据的存储无关,即独立于计算机,而存储结构是依赖于计算机的。计算机存储器是由有限多个存储单元组成的,每个存储单元有唯一的地址,各存储单元的地址是连续编码的,每个存储单元 z 都有唯一的后续单元 $z' = \text{succ}(z)$, z 和 z' 称为相邻单元。一片相邻的存储单元的整体叫做存储区域,记作 M 。把 B 存储在计算机中,首先必须建立一个从 K 的结点到 M 的单元的映象 $s: K \rightarrow M$, 即对于每一个 $k \in K$, 都有唯一的 $Z \in M$ 使得 $s(k) = z$, z 为 K 中结点所占存储空间中的起始单元。

数据的存储结构有顺序存储或链式存储两种方法。

顺序存储方法是把逻辑上相邻的元素存储在物理位置相邻的存储单元中,由此得到的存储表示称为顺序存储结构。顺序存储结构是一种最基本的存储表示方法,通常借助于程序设计语言中的数组来实现。其优点是占用最少的存储空间;其缺点是由于每个结点只能

使用一整块存储区域,因此可能产生较多的碎片现象,另外,在作插入或删除操作时需移动大量元素。

链式存储方法对逻辑上相邻的元素不要求其物理位置相邻,元素间的逻辑关系通过附设的指针字段来表示,由此得到的存储表示称为链式存储结构,链式存储结构通常借助于程序设计语言中的指针类型来实现。其优点是充分利用所有的存储单元,不会出现碎片现象;其缺点是每个结点占用较多的存储空间。

除了通常采用的顺序存储方法和链式存储方法外,有时为了查找方便还采用索引存储方法和散列存储方法。

索引方式是用结点的索引号来确定结点存储地址,其优点是检索速度快,其缺点是增加了额外的索引表,会占用较多的存储空间。另外,在增加和删除数据时还要修改索引表,因而会花费较多时间。

散列方式是根据结点的值确定它的存储地址,其优点是检索、增加和删除结点的操作都很快,其缺点是采用不好的散列函数时可能出现结点存储单元的冲突,为解决冲突需要另外的时间和空间开销。

七、数据类型

数据类型(Data Type)是一个值的集合和定义在这个值集上的一组操作的总称。

数据类型是和数据结构密切相关的一个概念。它最早出现在高级程序设计语言中,用以刻画程序中操作对象的特性。在用高级语言编写的程序中,每个变量、常量或表达式都有一个它所属的确定的数据类型。类型显式地或隐含地规定了在程序执行期间变量或表达式所有可能的取值范围,以及在这些值上允许进行的操作。

从硬件的角度来看,它们的实现涉及“字”、“字节”、“位”、“位运算”等等;从用户的观点来看,并不需要了解整数在计算机内是如何表示、运算细节是如何实现的,用户只需要了解整数运算的外部运算符性,而不必了解机器内部位运算的细节,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,用户只需注重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。

在高级程序设计语言中,数据类型可分为两类:一类是原子类型,另一类是结构类型。原子类型的值是不可分解的,如C语言中整型、字符型、浮点型、双精度型等基本类型,分别用保留字int、char、float、double标识;而结构类型的值是由若干成分按某种结构组成的,因此是可分解的,并且它的成分可以是非结构的,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。在某种意义上,数据结构可以看成“一组具有相同结构的值”,而数据类型则可被看成由一种数据结构和定义在其上的一组操作所组成的。

八、抽象数据类型

抽象数据类型(Abstract Data Type,简称ADT)是指一个数学模型以及定义在该模型上

的一组操作。抽象数据类型的定义取决于它的一组逻辑特性,而与其在计算机内部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,都不影响其外部的使用。

抽象数据类型和数据类型实质上是一个概念。例如,各种计算机都拥有的整数类型就是一个抽象数据类型,尽管它们在不同处理器上的实现方法可以不同,但由于其定义的数学特性相同,在用户看来都是相同的。因此,“抽象”的意义在于数据类型的数学抽象特性。

抽象数据类型的定义可以由一种数据结构和定义在其上的一组操作组成,本身是独立于实现的。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。而数据结构又包括数据元素及元素间的关系,因此抽象数据类型一般可以由元素、关系及操作三种要素来定义。

抽象数据类型的特征是使用与实现相分离,实行封装和信息隐蔽。就是说,在抽象数据类型设计时,把类型的定义与其实现分离开来。

另一方面,抽象数据类型的含义更广,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。所定义的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。从使用者的角度来看,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。

为了提高软件的重用性,在近代程序设计方法学中,要求在构成软件系统的每个相对独立的模块上,定义一组数据和施于这些数据上的一组操作,并在模块的内部给出这些数据的表示及其操作的细节,而在模块的外部使用的只是抽象的数据及抽象的操作。这也就是面向对象的程序设计方法。

无论是从计算机理论还是从计算机工程的角度来看,抽象数据类型的概念都是十分重要的。一方面,它抽象和推广了高级程序设计语言(例如 C 语言)中的类型概念;另一方面,也为软件工程提供了一个自顶向下的实现图式。

用抽象数据类型的概念来指导问题求解的过程,可以用图 1-6 来表示。

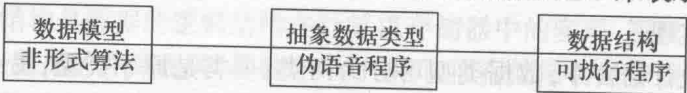


图 1-6 用抽象数据类型指导问题求解

第四节 算法和算法分析

算法与数据结构的关系紧密,在算法设计时先要确定相应的数据结构,而在讨论某一种数据结构时也必然会涉及相应的算法。描述一个算法可以采用某一种计算机语言,也可以采用流程图等。下面就从算法特性、算法描述和算法性能分析与度量三个方面对算法进行介绍。

