

全国高等院校计算机基础教育研究会
“计算机系统能力培养教学研究与改革课题”立项项目

数据结构

(用C++语言描述)

吴艳 赵端阳 曹平 等◎编著

图解详尽 | 浅显易懂，抽象到具体，分析过程一目了然

案例教学 | 浅入深出，支持高效的代码重用，给出富于启发性的问题和讨论

注重实践 | 循序渐进，设计典型的习题和实验指导，突出技能提升



北京邮电大学出版社
www.buptpress.com

全国高等院校计算机基础教育研究会

“计算机系统能力培养教学研究与改革课题”立项项目

数据结构

(用C++语言描述)

吴艳 赵端阳 曹平 等◎编著



北京邮电大学出版社
www.buptpress.com

内 容 简 介

“数据结构”是计算机专业的核心课程,是从事计算机软件开发和应用人员必修的专业基础课。随着计算机学科的迅速发展,“数据结构”课程也在不断增加新的内容,在不断发展。

本书采用能够自然体现抽象数据类型概念的 C++ 语言作为算法描述语言,从线性结构到非线性结构,从简单到复杂,深入描述了各种数据结构内在的逻辑关系及其在计算机中的实现方式和具体的应用。全书的内容包括线性表、栈、队列、数组、串、广义表、树、图、查找以及各种排序方法。此外,对常用的迭代、递归、回溯以及贪心等算法设计技巧,搜索和排序算法做了详尽的描述,并引入了一些比较高级的数据结构和简单的算法分析。

本书可作为高等院校计算机专业、软件专业的本科生教材,同时也可作为准备参加研究生入学考试、自学考试和各类程序设计竞赛人员的参考书。

图书在版编目(CIP)数据

数据结构:用 C++ 语言描述 / 吴艳等编著. -- 北京:北京邮电大学出版社, 2016.5

ISBN 978-7-5635-4733-3

I. ①数… II. ①吴… III. ①数据结构②C 语言—程序设计 IV. ①TP311.12②TP312

中国版本图书馆 CIP 数据核字 (2016) 第 071909 号

书 名: 数据结构 (用 C++ 语言描述)

著作责任者: 吴 艳 赵端阳 曹平等 编著

责任编辑: 王丹丹 刘 佳

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号 (邮编: 100876)

发行部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京鑫丰华彩印有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 19

字 数: 482 千字

印 数: 1—2 000 册

版 次: 2016 年 5 月第 1 版 2016 年 5 月第 1 次印刷

ISBN 978-7-5635-4733-3

定 价: 38.00 元

• 如有印装质量问题, 请与北京邮电大学出版社发行部联系 •

前 言

随着计算机的普及,人们对软件的需求量日益增长。为了提高软件的生产力,降低软件的开发成本,以高质量来满足各种合乎需要的软件,必须遵循软件工程的基本原则,把软件的开发和维护过程标准化、工程化。就软件产品而言,最重要的就是建立合理的软件体系结构和软件结构,设计有效的数据结构。因此,从事软件开发必须要了解各种数据的表示、关联和处理方法,以及如何组织数据在计算机中的存储、传递以及转换,而这些正是数据结构课程的主要内容。

“数据结构”作为一门独立的课程在国外是从 1968 年才开始设立的。1968 年美国 D. E. 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)》是第一部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。“数据结构”在计算机科学中是一门综合性的专业基础课,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。“数据结构”这一门课的内容不仅是一般程序设计(特别是非数值性程序设计)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的重要基础。

“数据结构”课程内容丰富、涉及面广泛,而且其内容也随着各种基于计算机的应用和应用技术的发展而拓宽和发展。面向对象软件设计方法出现以后,各种数据结构的讨论都是基于抽象数据类型和软件复用的,更加符合人们认识自然、理解自然的习惯,开发出来的软件系统质量容易得到保证,且易于理解、扩充和维护。

编制本书的目的是为了引导学生在掌握课程的基本内容基础上,了解常见的数据结构的实际应用,举一反三,循序渐进,具备分析问题能力、算法设计能力,最终具备解决复杂问题的能力,从而提高软件设计和编程能力。

为此,我们结合当前与课程内容紧密相关的实际应用,参考了众多的数据结构教材,在教学内容选择、教学步骤设计、习题的编排以及实验文档组织等诸方面都做了精心的考虑和安排,邀请长期工作在教学第一线且年富力强的各课程资深专业教师,编写了这本“数据结构”教材。书中采用了自然体现数据抽象思想的 C++ 程序设计语言作为算法描述语言,内容覆盖了“数据结构”课程的教学大纲,课程涉及的数据结构主要是一些重要的、应用最广泛的结构,包括线性表、栈、队列、数组、串、广义表、树、图以及集合。本书对这些结构做了全面的讲解,阐明了各种数据结构内在的逻辑关系,讨论了它们在计算机中的存储表示以及存储特点,并结合各种典型实例说明它们在解决实际问题时的动态行为和各种必要的操作,并通过使用 C++ 程序设计语言介绍用面向对象技术表达和实现各种数据结构。

本教材的编写原则是:

(1) 作为数据结构入门教材,结合具体的实例使学生能较快理解各种数据结构内在的逻辑关系,掌握具体数据结构的表示;

(2) 介绍各种数据结构在计算机内的存储表示,并对比不同存储表示的特点、主要区别以及适用的场合;

(3) 针对不同的存储表示,用 C++ 编程语言写出相应的类定义,当涉及多个类时,还必须确定类与类之间的关系;

(4) 实现各种数据结构中的基本操作。

本教材的特色主要体现在以下几个方面:

(1) 借助示意图描述数据结构内在的逻辑关系、数据结构在计算机内的存储表示、算法描述以及具体实例的运行结果,简洁明了,便于理解;

(2) 结合实际应用设计具体实例,使得抽象的内容变得具体、浅显易懂;

(3) 大多数算法使用了泛型编程,支持高效的代码重用;

(4) 设计了许多有典型性的习题(难度随着题号增大而增大)和实验,实验中包含一些教学反馈信息,有助于教师了解学生学习情况;

(5) 针对重点算法给出启发式的提示和探讨。

全书采用 C++ 语言作为数据结构与算法的描述语言,通过一系列教学、实验以及练习,把数据结构的概念、理论知识与技术融入实际应用中去,从而加深对本课程的认识和理解,进一步熟悉和掌握面向对象的程序设计方法,提高编程能力和综合分析能力,并为后续课程的学习打下扎实的基础。

感谢周苏教授在本书形成过程中提供的诸多帮助和宝贵建议!本书的编写得到了浙江工业大学之江学院、浙江大学城市学院等多所院校领导及师生的大力支持,在此一并表示感谢!感谢北京邮电大学出版社编辑刘佳女士等为本书的出版所做的大量工作!

编者

2016 年 3 月于西子湖畔

读者指南

任何实际问题只有建立了数学模型才可以被计算机计算,而数据结构就是实际问题中元素的数学抽象,算法则是建立和解决数学模型的方法。

“数据结构”这个术语在整个计算机科学与技术领域中被广泛使用,它被用来反映一个数据的内部构成,即一个数据由哪些成分数据构成,以什么方式构成,呈什么样的结构等。数据结构是数据存在的形式,也是信息的一种组织方式,其目的是为了提高算法的效率,它通常与一组算法的集合相对应,通过这组算法集合可以对数据结构中的数据进行某种操作。

“数据结构”是计算机等相关专业的一门核心课程,具有承上启下的地位和作用,“程序设计语言”(例如 C、C++ 或 Java) 和“计算机组成”是它的先导课程,“操作系统”“数据库原理”“软件工程”等是它的后续课程。

全书采用 C++ 语言作为数据结构与算法的描述语言,通过一系列实例、实验和练习,把数据结构的概念、理论知识与技术融入实际应用中去,从而加深对本课程的认识和理解,进一步熟悉和掌握面向对象程序设计方法,提高编程能力和综合分析能力,并为后续课程的学习作一些铺垫。

本书中的所有程序都通过了 Microsoft Visual C++ 6.0 软件开发环境下的调试运行,以尽可能地保证所给出算法和程序的正确性和有效性。

读者对象

高等院校计算机和信息管理等相关专业“数据结构”课程的学生可把此书作为主教材使用。对于已经具备初步的计算机应用和程序设计知识,并希望通过进一步学习得到提高、希望通过计算机等级考试的读者来说,本书也是一本自学和继续教育的良好读物。

本书的实验内容有助于“数据结构”课程的教与学,有助于读者对掌握和理解本课程内容建立起足够的信心和兴趣。

教学内容

本书的教学内容、实验以及练习几乎覆盖了“数据结构”课程教学的各个方面,内容涉及数据结构和算法分析基础、线性表、栈和队列、串、树和二叉树、图以及查找与内部排序等,全书共 9 章,9 个实验练习和 1 个实验总结。实验练习的难易程度不同,以帮助读者加深对教材中概念的理解,并逐步引导学生对本课程的深入学习。

第 1 章:绪论。包括数据结构的概念、抽象数据类型的表示和实现以及算法和算法分析等三部分内容。理解抽象数据类型的特点、定义方法和在 C++ 语言环境下实现的方法;掌

握算法的主要特征和描述方法；尝试通过具体的算法结构，计算算法的时间复杂度和空间复杂度，并对算法进行定性或定量评价。

第 2 章：线性表。包括线性表的逻辑结构、线性表的顺序表示和实现、线性表的链式表示和实现以及链表和顺序表的选取等四部分内容。理解并掌握顺序表和链表的存储结构定义；实现基本操作算法的描述和分析；进一步理解实现数据结构的 C++ 类的定义、模板类与泛型编程，从而更深入理解面向对象程序设计的方法；与此同时，能较合理地在实际应用中选取存储结构。

第 3 章：栈和队列。包括栈、队列以及栈和队列应用等三部分内容。理解并掌握顺序栈和链式栈、顺序队列和链式队列的类定义；掌握栈和队列基本操作的过程及实现的方法；理解栈和队列作为辅助结构求解问题的思路、使用条件以及设计方法；进一步加深对栈和队列特点的理解，并区别这两种结构在解决实际问题时的区别，从而能在实际应用中合理地选择适当的数据结构。

第 4 章：广义表和数组。包括广义表和数组两部分内容。理解数据元素在广义表和数组中的逻辑关系；掌握广义表和数组的基本操作；理解广义表和数组的不同存储方式以及在具体存储方式下基本操作的实现；进一步体会存储结构对算法效率的影响。

第 5 章：串。理解并掌握串的三种常见的存储结构，以及在具体的存储结构下串的基本操作和实现。加深理解串这种常见的数据结构的特点和存储结构定义方法；掌握串的基本操作算法的描述，并能对算法进行评价和选取；理解串操作的实现方法，理解一个合理的存储结构定义对具体操作实现的重要性，并注重提高算法的健壮性，从而能更好地理解一个好算法所需要的各种综合因素。

第 6 章：树和二叉树。包括树的定义及表示、二叉树、二叉树遍历及其应用、线索二叉树、树和森林和 Huffman 树以及其应用等六部分内容。掌握二叉树的不同存储结构，并理解在具体的应用中采用合理的存储结构的思路；加深理解递归算法的设计思路，体会栈在递归算法中的作用；理解广度搜索算法的设计思路，体会队列在实际应用中的作用；理解 Huffman 树的特点、存储结构的选取以及建立 Huffman 树建立的算法思路，并在此弄清哈夫曼编码设计过程和实现代码的程序结构，从而加深对 C++ 语言中字符串和指针的理解和灵活运用。

第 7 章：图。包括图的基本概念、图的存储结构、图的遍历、最小生成树、最短路径表示和 DAG 及其应用等六部分内容。掌握图的各种存储结构，从而更进一步理解如何根据实际问题设计合理的模板类；掌握图的两种遍历方法，即图的深度优先搜索和广度优先搜索，并根据这两种操作的实现过程确定算法中所需要采用的辅助数据结构；能根据实际问题选择图的类型和合理的存储结构，理解解决最小生成树、拓扑排序、关键路径和最短路径等经典算法的设计思想，并能根据算法写出相应的程序代码。根据贪心算法的核心思想理解贪心算法的特点，并学会设计简单的贪心算法。

第 8 章：查找。包括查找的基本概念、静态查找、树表的查找以及散列表查找等四部分内容。掌握查找表的特点，并能根据具体的查找方法定义合理的查找表的存储结构；掌握各种查找表的实现思路，能根据查找思路写出算法的描述，并对算法进行评价；掌握哈希查找的特点和影响哈希查找的因素，从而构造合理的哈希函数、采用适当的解决冲突的方法，加深理解 C++ 语言中指向函数的指针的具体应用。

第9章:内部排序。包括排序的概念及算法性能分析、插入排序、交换排序、选择排序、归并排序以及基数排序等六部分内容。熟悉各种排序方法的算法思路,并能根据具体的实现方法定义合理的待排序数据的存储结构;掌握各种排序方法的实现思路,能根据排序思路写出算法的描述,并对相应算法进行时间复杂度和空间复杂度计算;体会各种排序算法在最好情况下和最坏情况下的算法评价,并能根据待排序数据的实际分布,确定相应的排序方法。

本书在每章之后都安排一次习题和实验,各个习题及实验难度有所不同,随着编号的增加难易程度也不断增加,从这个意义上讲,应该在完成前面部分的相关知识和习题之后再行后面实验。如果在做实验时遇到了困难,你可能需要搜索更早的实验练习来帮助解决问题。

尽管在各章内容中涉及许多C++相关知识,但本书不会对这些C++内容深入说明,因此,在学习本课程的同时,学习相关的“C++语言程序设计”课程,则可以从实验和练习中获取更多的知识,来提高自己的综合分析能力和编程能力。

实验要求

根据不同的教学安排和要求,课程中的实验学时数也有所不同。

致教师

数据结构与算法的应用面广,涉及技术领域宽泛,也被人们赋予了很高的期望值。另一方面,要学好“数据结构”课程,仅仅通过课堂教学或自学获取理论知识是远远不够的。因此,要让学生真正理解数据结构与算法的基础理论知识,具备将数据结构与算法知识应用于社会实践的能力,积极加强数据结构课程的实验环节是至关重要的。

本书结合软件图形描述工具和相应的高级语言开发工具,通过提供一组与单元知识密切相关的实验练习作为对理论知识的补充,有助于学生对理论知识的理解,有助于提供学生的应用和开发能力。

为方便教师对课程实验的组织,我们在实验内容的选择、实验步骤的设计和实验文档的组织等诸方面都做了精心的考虑和安排。任课教师不需要投入很多精力来设计实验练习。相反,教师和学生都可以通过本书提供的实验练习来理解概念和实现应用。任课教师需要在开头的几个实验中引导学生熟悉实践环境、理解抽象的数据类型的定义方法和体会自顶向下的结构化程序设计风格,使学生能按照实验步骤循序渐进地进行数据结构设计、算法设计和程序的调试。

本书的全部实验都经过了教学实践的检验,取得了较好的教学效果,大部分学生在这种实验模式下进行实践后,分析能力和编程能力得到了显著提高。但是,在实验过程中学生仍普遍存在以下几个问题:

(1) 实验前的准备工作不充分。学生常常会忽视对每个实验的相关知识的理解和回顾,而一味只求完成实验步骤。

(2) 实验中不注重理解所做的实验内容。只是按部就班地照着实验步骤进行程序代码的输入和调试,遇到错误提示,不会积极主动地进行排错。

(3) 实验步骤完成后不能及时进行总结。为了赶时间,往往草草了事,没有投入时间对刚完成的实验内容进行消化,需要反复多次才能熟悉原本较为容易掌握的知识内容。

因此,为了保证实验的质量,建议教师重视对教学实验环节的组织,例如:

(1) 要求学生对实验内容进行预习,并把预习重点放在实验中涉及相关课程知识。实验指导老师在实验开始初期对学生的预习情况进行检查,计入实验成绩。

(2) 明确要求学生重视对实验内容的体会和理解,认真完成“实验总结”,并把这部分内容作为实验成绩的主要评价成分,以促使学生对所学知识的理解,并做到举一反三。

(3) 对于有条件的学校(例如,学生自配有电脑,或实验室有足够的课余上机时间安排的),不理解的实验部分可以建议学生反复实践,以加深对核心和重要内容的理解。

如果需要,教师可以根据学生的学习情况,在现有实验的基础上,结合当前热门的实际应用给出一些要求、指导和布置,以进一步发挥学生的潜能和激发学生学习的主动性和创造性。

致学生

对于计算机及其相关专业的学生来说,数据结构肯定是需要掌握的重要专业基础知识之一。但是,单凭课堂教学和一般作业,要真正领会数据结构课程所介绍的概念、原理、方法和技巧等,是很困难的,应该通过大量的上机实验,才能真正理解相关知识,并做到灵活运用。

另一方面,经验表明,学习尤其是真正体会和掌握数据结构与算法的最好方式是对它进行充分的实践,无疑,通过了解、熟悉和掌握数据结构与算法程序设计,是应用数据结构与算法知识的重要途径。学生必须通过了解、熟悉一些经典算法,来逐步掌握一些编程技巧,提高自己的分析和判断能力。

实验设备

个人计算机在学生,尤其是专业学生中的普及,使得我们有机会把实验任务分别利用课内和课外时间来完成,以获得更多的锻炼。这样,对实验室和个人计算机的配置就有不同的要求。

实验室设备与环境

用来进行本书数据结构实验的实验室环境,大都需要其计算机设备安装有 Borland C++ 5.0 或 Microsoft Visual C++ 6.0 或 Visual studio 2015 开发环境。

由于部分实验有可能无法一次完成,有些实验在内容上有一定的互通性和连贯性,所以,实验室设备应能帮助并注意提醒学生妥善保存其实验内容。

个人实验设备与环境

用于本书数据结构实验的个人计算机环境,一般建议在其 Windows 2000 Professional、Windows XP Professional 操作系统中安装有 Visual Studio 2015 或 Microsoft Visual C++ 6.0 开发环境。个人计算机环境需要为实验准备足够的硬盘存储空间,以方便实验软件的安装和实验数据的保存。

在利用个人计算机完成实验时,要重视理解在操作中系统所显示的提示甚至警告信息,注意保护自己的数据和计算环境的安全,做好必要的数据备份工作,以免产生不必要的损失。

由于有些实验在内容上有一定的互通性和连贯性,所以,要注意妥善保存自己的实验内容。



目 录

第 1 章 绪论	1
1.1 数据结构的概念	1
1.1.1 为什么要学习数据结构	3
1.1.2 数据结构主要研究的内容	4
1.2 抽象数据类型的表示和实现	6
1.2.1 数据类型	6
1.2.2 抽象数据类型	7
1.2.3 抽象数据类型表示	8
1.2.4 抽象数据类型实现	8
1.3 算法和算法分析	10
1.3.1 算法定义	10
1.3.2 算法描述	11
1.3.3 算法性能分析与度量	12
1.3.4 常见的算法类型	15
本章总结	15
练习	15
实验 1	18
第 2 章 线性表	20
2.1 线性表的逻辑结构	20
2.1.1 线性表的定义	20
2.1.2 线性表的抽象数据类型定义	20
2.2 线性表的顺序表示和实现	21
2.2.1 线性表的顺序表示	21
2.2.2 顺序表表示	22
2.2.3 顺序表基本操作的实现	23
2.2.4 顺序表应用举例	27
2.3 线性表的链式表示和实现	28
2.3.1 单向链表的概念	29
2.3.2 链表的类定义	31
2.3.3 链表基本操作的实现	32
2.3.4 双向链表	36

2.3.5 链表应用实例	39
2.4 链表和顺序表的选取	41
本章总结	42
练习	43
实验 2	46
第 3 章 栈和队列	49
3.1 栈	49
3.1.1 栈的定义及操作	49
3.1.2 栈的抽象数据类型定义	50
3.1.3 栈的存储及操作实现	50
3.1.4 栈的应用	56
3.2 栈与递归	63
3.2.1 递归的概念	64
3.2.2 递归过程与递归工作栈	66
3.2.3 递归算法向非递归算法的转换	67
3.2.4 递归的应用	69
3.3 队列	70
3.3.1 队列的定义及基本操作	70
3.3.2 队列的抽象数据类型	70
3.3.3 队列的存储及操作实现	70
3.3.4 双端队列	75
3.3.5 队列的应用	75
本章总结	76
练习	76
实验 3	79
第 4 章 数组和广义表	82
4.1 数组	82
4.1.1 多维数组的概念与存储表示	82
4.1.2 特殊矩阵及压缩存储	84
4.2 稀疏矩阵的压缩存储	85
4.2.1 稀疏矩阵的三元组表示	86
4.2.2 稀疏矩阵的链式存储法	95
4.3 广义表	97
4.3.1 广义表的基本概念	97
4.3.2 广义表的存储结构	99
本章总结	105
练习	105
实验 4	108

第5章 串	110
5.1 串的基本概念及抽象数据类型	110
5.1.1 串的基本概念	110
5.1.2 串的抽象数据类型	110
5.1.3 C++有关串的库函数	111
5.1.4 串的存储结构	113
5.2 串的顺序存储结构及基本操作实现	114
5.2.1 串的顺序存储结构	114
5.2.2 串的基本操作及实现	115
5.2.3 串的模式匹配	119
5.3 串的链式存储	123
本章总结	124
练习	124
实验5	126
第6章 树和二叉树	129
6.1 树的定义及表示	129
6.1.1 树的定义	129
6.1.2 树的表示	130
6.2 二叉树	131
6.2.1 二叉树的定义	131
6.2.2 二叉树的抽象数据类型	134
6.2.3 二叉树的存储结构	134
6.2.4 二叉树结点类操作实现	136
6.2.5 二叉树类操作实现	137
6.3 二叉树遍历及其应用	138
6.3.1 二叉树遍历的递归算法	138
6.3.2 二叉树遍历的应用	140
6.3.3 二叉树遍历的非递归算法	142
6.4 线索二叉树	144
6.4.1 线索二叉树定义	144
6.4.2 线索二叉树存储结构	145
6.4.3 线索二叉树基本操作	146
6.5 树和森林	149
6.5.1 树的存储表示	149
6.5.2 树和森林的遍历	155
6.6 Huffman树及其应用	157
6.6.1 最优二叉树概念	157
6.6.2 最优二叉树的构造	158

6.6.3 Huffman 树的应用:Huffman 编码	162
本章总结	164
练习	165
实验 6	167
第 7 章 图	170
7.1 图的基本概念	170
7.1.1 图的定义和术语	170
7.1.2 图的抽象数据类型	172
7.2 图的存储结构	172
7.2.1 图的邻接矩阵表示	173
7.2.2 图的邻接表表示	179
7.2.3 图的十字链表表示	187
7.2.4 图的邻接多重表表示	188
7.3 图的遍历	188
7.3.1 深度优先搜索	189
7.3.2 广度优先搜索	190
7.4 最小生成树	191
7.4.1 Prim 算法	192
7.4.2 Kruskal 算法	195
7.5 最短路径	197
7.5.1 路径的概念	197
7.5.2 从一个顶点到其余各顶点的最短路径	197
7.5.3 每对顶点之间的最短路径	200
7.6 DAG 及其应用	203
7.6.1 AOV 网络与拓扑排序	203
7.6.2 AOE 网络与关键路径	206
本章总结	209
练习	210
实验 7	213
第 8 章 查找	216
8.1 查找的基本概念	216
8.2 静态查找表	217
8.2.1 顺序查找	218
8.2.2 二分查找	219
8.2.3 分块查找	221
8.3 树表的查找	223
8.3.1 二叉排序树	223
8.3.2 平衡二叉树	227

8.3.3 红黑树	230
8.3.4 B 树	235
8.3.5 B ⁺ 树	238
8.4 散列表查找	239
8.4.1 散列表的基本概念	239
8.4.2 哈希函数的构造方法	240
8.4.3 处理冲突方法	241
本章总结	245
练习	246
实验 8	249
第 9 章 内部排序	252
9.1 排序的概念及算法性能分析	252
9.1.1 排序的概念	252
9.1.2 排序算法的性能分析	253
9.1.3 排序表类定义	253
9.2 插入排序	255
9.2.1 直接插入排序	255
9.2.2 希尔排序	257
9.3 交换排序	258
9.3.1 冒泡排序	259
9.3.2 快速排序	260
9.4 选择排序	263
9.4.1 直接选择排序	263
9.4.2 堆排序	265
9.5 归并排序	271
9.5.1 归并	271
9.5.2 归并排序算法	273
9.6 基数排序	273
9.6.1 基数排序思想	273
9.6.2 LSD 基数排序	274
9.7 各种内部排序方法比较	277
本章总结	278
练习	279
实验 9	282
附录 实验总结	285
参考文献	288



第 1 章 绪 论

学而不思则罔,思而不学则殆。

学习目标

- 掌握数据结构的基本概念,理解数据的逻辑结构与存储结构之间的区别与联系。
- 明确抽象数据类型的定义、表示以及实现方式。
- 理解抽象数据类型与 C++ 类的对应关系,从而理解如何用 C++ 语言描述数据结构。
- 明确算法的作用,理解并掌握算法时间复杂度和空间复杂度的估算方法。

言简意赅地说,计算机科学是一门研究数据表示和数据处理的科学。数据是计算机可以直接处理的最基本和最重要的对象。对于将要从事计算机系统开发的专业人员来说,数据结构是一个门槛,是必须切实掌握的知识;同时,它还是继续开展计算机业务的技术基础。

1.1 数据结构的概念

除了进行科学计算之外,计算机已被广泛地应用在控制、管理和数据处理等非数值计算的领域中。与此相应,处理对象也由早先纯粹的数值发展到字符、表格和图形图像等各种具有一定结构的数据,这给计算机程序设计带来了新的问题。为了编写一个“好”的程序必须明确处理对象的特征及各对象之间的关系。这就是“数据结构”这门学科形成和发展的背景。

“数据结构”作为一门独立的课程,在 1968 年由美国计算机科学家 D. E. 克努特教授首先开创的,他所著的《计算机程序设计技巧(第一卷基本算法)》是第一本较系统阐述数据的逻辑结构和存储结构及其操作的著作。

数据结构作为计算机科学的一门分支科学,主要研究非数值计算的程序设计问题中计算机的操作对象、对象之间的关系和操作等。

例如,假设要设计一个电话号码簿系统。

(1) 确定电话号码信息(对象)之间的关系,即电话号码簿实际上是多个电话号码信息组成的一个序列(对象之间的关系或逻辑数据结构)。在具体的问题求解分析过程中得到一个对象的过程就是抽象。

(2) 根据电话号码簿在日常生活中的实际应用,确定对它进行的一系列操作(电话号码信息对象的操作)及实现步骤(算法),如添加、修改或者删除一个或多个电话号码信息、查找特定的电话号码信息以及打印号码等,以及这些操作的具体实现步骤。

(3) 将(1)确定的逻辑关系或逻辑数据结构采用一定的方法存储在计算机中(逻辑数据结构在计算机中表示,即存储结构),并用具体的程序设计语言实现(2)中确定的操作。操作的实现步骤序列,简称为算法,算法是独立于具体的编程语言的,可以用不同的程序设计语言实现同一个算法。

(4) 需要评价设计中使用的逻辑数据结构、存储结构以及算法。在实际操作中往往需要按照不同的策略选择不同的解决方案。例如,可按照姓氏笔画或建立时间顺序存储信息;也可按电话号码顺序查找指定电话信息;还可建立号码-页码索引进行号码搜索。可以从不同的角度评价解决方案的合理性和有效性。

数据结构课程重在讨论软件开发过程中的方案分析和设计阶段、编码分析和设计阶段上的若干基本问题。此外,构造好的数据结构及其实现,还要考虑数据结构的选择与评价。因此,数据结构内容包括三个层次的两个方面的内容,如表 1-1 所示。

表 1-1 数据结构内容体系

方面 层次	数据表示	数据处理
抽象	逻辑数据结构	基本操作
实现	存储结构	算法
评价	不同数据结构的比较及算法分析	

可以得出数据结构的三要素为数据的逻辑结构、数据的存储结构以及数据的操作。

- (1) 通过抽象层,舍弃数据元素的具体内容,就得到逻辑结构的表示;
- (2) 通过分解和抽象将处理要求划分成各种功能、舍弃实现细节,得到操作的定义;
- (3) 将问题转化为数据结构的过程即是一个从具体(具体问题)到抽象(数据结构)的过程;然后,通过增加对细节实现的考虑得到存储结构和具体操作(用算法描述),即实现从一个抽象(数据结构)到具体(操作)的过程。

“数据结构”是计算机专业基础课,它与其他课程之间的衔接关系如图 1-1 所示。

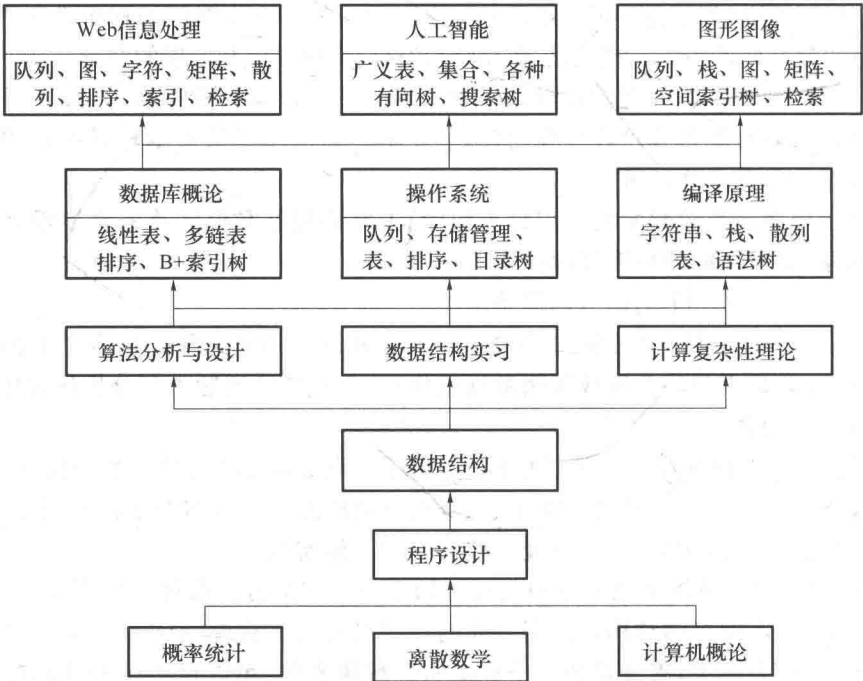


图 1-1 “数据结构”在计算机专业课程群中的关键地位

任何实际问题只有建立了数学模型才可以被计算机操作,而数据结构就是实际问题中操作对象(数据元素)的数学抽象,算法则是建立和解决数学模型的方法。

1.1.1 为什么要学习数据结构

使用计算机解决具体问题一般需要经过以下几个步骤:

- (1) 从具体问题抽象出适当的数学模型;
- (2) 设计或选择解决此数学模型的算法;
- (3) 编写程序并进行调试、测试,直至得到最终的解。

随着计算机应用领域的扩大和软硬件的发展,非数值计算问题显得越来越重要。这类问题使得描述问题的数据元素之间的相互关系一般无法用数学方程式来表示。因此,解决问题的关键不再是数学分析和计算方法,而是设计出合理的数据结构。

程序设计的实质是对实际问题进行设计、选择合适的数据结构和高效率的算法。而高效的算法在很大程度上取决于描述实际问题的数据结构。

【例 1-1】 图书检索问题。如果要在图书馆搜索图书的相关信息,可以输入书名或出版社找到相关的一个或几个图书分类号(同名的不同图书)。可以根据图书分类号得到需要的图书的相关信息,如作者、出版社、内容提要以及目录等,如图 1-2 所示的存储结构能实现图书检索。在类似的信息管理系统的数学模型中,计算机处理的对象之间的关系存在着一种简单的线性关系。这类数学模型可以称为线性结构。

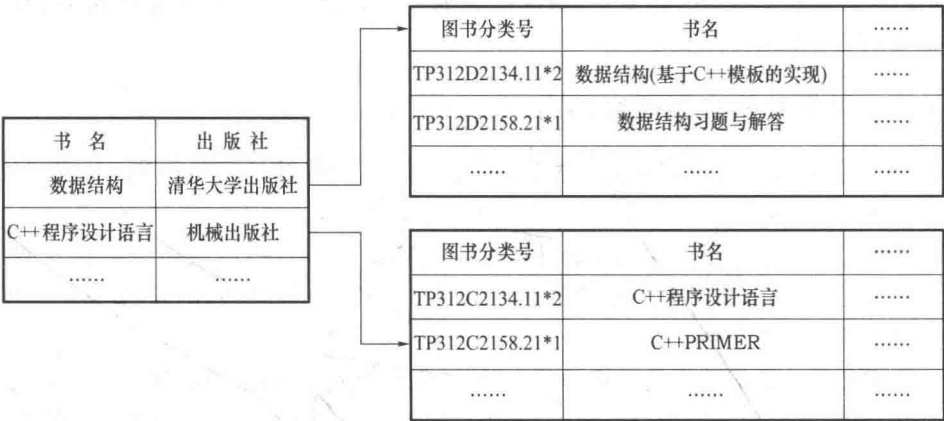


图 1-2 图书的检索结构

【例 1-2】 搜索指定文件夹问题。从根目录开始依次查找它的各个子目录,如果查到某个子目录最底层目录还未查找到指定文件夹,则返回上层目录,寻找它的另一个子目录;依此类推,直到找到指定的文件夹或搜索失败为止。在搜索文件夹的过程形成了一棵隐含的状态树(如图 1-3 所示),树也是一种常见的数据结构。

【例 1-3】 铺设公园景点石板通道问题。公园的 N 个景点之间有多条石板通道,要求有 $N-1$ 条通道将所有的 N 个景点连接,并且所有通道的造价之和要最小。首先,将问题转化为一个图形模型:图中顶点代表景点,边代表可行的石板通道;然后,将问题求解转化为求解最小