


高等学校计算机专业规划教材




数据结构与算法

C语言描述

第2版

沈华 文志诚 杨晓艳 张明武 编著



*Data Structures
and Algorithms in C
Second Edition*



机械工业出版社
China Machine Press

高等学校计算机专业规划教材

数据结构与算法

C语言描述

第2版

沈华 文志诚 杨晓艳 张明武 编著

*Data Structures
and Algorithms in C
Second Edition*

 机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

数据结构与算法: C 语言描述 / 沈华等编著. —2 版. —北京: 机械工业出版社, 2015.8
(高等学校计算机专业规划教材)

ISBN 978-7-111-51142-7

I. 数… II. 沈… III. ① 数据结构-高等学校-教材 ② 算法分析-高等学校-教材 ③ C 语言-程序设计-高等学校-教材 IV. ① TP311.12 ② TP312

中国版本图书馆 CIP 数据核字 (2015) 第 191863 号

本书运用实例法、图示法和问题驱动法等教学方法生动、系统地介绍各种常用的数据结构以及排序、查找算法, 阐述各种数据结构的逻辑关系、存储表示及运算。全书共分为四个部分: 第一部分主要介绍什么是数据结构, 什么是算法, 它们之间有着怎样的联系, 以及如何进行算法分析。第二部分和第三部分分别重点介绍常见的线性结构和非线性结构。在实际应用中, 最常遇到的两个运算是查找 (即搜索) 和排序, 实现这两种运算的各种算法将在第四部分介绍。

本书可以作为高等院校计算机科学与技术及相关专业本科生的教材, 也可以作为报考高等学校计算机专业硕士研究生入学考试的复习用书, 同时还可以作为广大工程技术人员的参考资料。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 余洁 朱劼

责任校对: 殷虹

印刷: 北京瑞德印刷有限公司

版次: 2015 年 10 月第 2 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 22.5

书号: ISBN 978-7-111-51142-7

定价: 45.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

前 序

在计算机科学中，数据结构是一种在计算机中组织和存储数据，以便高效利用这些数据的有效方式。几乎所有的程序或软件系统都用到了数据结构。数据结构是许多高效算法的基本要素，同时它也使得管理大规模数据成为可能。数据结构课程是计算机科学的一门非常重要的专业基础课，也是 IT 类各专业的核心基础课。

数据结构中的数据和结构是两个紧密联系而又相互关联的概念，数据是数据结构的主要组成部分，它不涉及数据之间的关系；结构才涉及而且只涉及数据之间的关系。从中文构词上来看，数据结构更强调的是结构，即数据之间的关联方式。我们需要将适用于计算机的问题求解策略用计算机能理解的形式输入计算机中，告诉计算机如何一步一步地处理数据并最终得到问题的解，这就是算法。因此，从抽象层面上讲，数据结构是为算法服务的。不同的应用需求需要不同的算法，不同的算法需要不同的数据结构来支持。从实现层面上讲，数据结构是为算法实现服务的。

本书作者具有多年的教学和科研经验，对各种数据结构及应用算法有深入的了解。本书对什么是数据结构、什么是算法、算法和数据结构之间的关系进行了生动的阐述，并对各种线性数据结构、非线性数据结构、查找算法和排序算法等做了详尽的描述和分析。本书基本概念清晰，重点和难点问题讨论深入，而且循序渐进，为读者深入理解和应用数据结构给出了启示。本书既注重对数据结构经典内容的论述，又强调数据结构的应用，对相关内容进行了启发式讨论，是一本值得阅读和选用的教科书。

何炎祥

武汉大学计算机学院

2010 年 8 月 18 日于武汉珞珈山

前 言

计算机在社会各领域的应用已经无处不在。计算机需要加工处理的数据越来越复杂、规模也越来越大，对数据组织和处理的效率提出了更高的要求。在计算机科学中，数据结构是一种在计算机中组织、存储数据，以便高效利用这些数据的有效方式，它是许多高效算法的基本要素，几乎所有的程序或软件系统都要用到数据结构。“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1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。数据结构课程是计算机科学的基础核心课程，它对于外围课程的知识与经验具有迁移性、衍生性，并对学生自学知识和获取实践经验具有基础性、发展性。该课程教学的突出难点是知识的抽象性和动态性，学习过程也是复杂程序设计的训练过程，理论性和实践性均较强，被认为是比较难学的课程。对学生来说，学习这门课程的主要难度体现在，不知道如何根据实际问题选择合适的数据结构（包括两个层次，如何选择合适的数据逻辑结构构建问题模型和如何选择合适的数据存储结构实现求解问题模型的算法）和如何应用选择的数据结构进行问题的求解（当然，这个问题还需要“算法设计与分析”这门课程来进一步解决）。

基于上述考虑，本书充分运用实例法深入浅出、形象地讲解各种数据结构，针对一个讲解实例给出了不同的求解方案，让读者能够了解如何选择合适数据结构求解问题，并亲身体会采用同一种数据结构的存储结构实现同一个算法在执行效率和编程难度上的差异，从而激发学生的学习兴趣 and 热情。而且，以问题驱动的方式引导学生进行逐步深入思考，有利于帮助学生建立计算思维，使得他们不仅知其然，而且知其所以然。

本书分为四个部分共12章。第一部分包括第1章和第2章，主要介绍数据结构的概 念、算法的概念、数据结构和算法之间的密切关联以及简单的算法分析。第二部分包括第3~7章，这一部分以“线性表”为主线分别详细讨论了六种常见的线性结构。第三部分介绍了非线性结构中树（第8章）、二叉树（第9章）和图（第10章）的基本概念、存储结构、一些基本操作的实现以及它们的经典应用。第四部分介绍了查找（第11章）和排序（第12章）的各种常用算法，同时展现了各种数据结构的完美应用。

本书在内容组织上力求丰富充实、科学合理，符合学生的认知规律，力求将每种数据结构分析透彻。在语言描述上力求深入浅出、简单明了。为了帮助学生理解和掌握各种数据结构，书中列举了大量的思考题、例题和习题。本书主要采用面向过程的C语言作为数据结构和算法的描述手段，在保持C语言优点的同时尽量使算法描述简单清晰。

与本书配套的资料有书中程序的源代码、习题参考答案以及电子课件等，教师可登录

华章网站下载。

本书第 1~10 章主要由湖北工业大学沈华博士编写,第 11~12 章主要由湖南工业大学文志诚博士编写,湖北工业大学杨晓艳博士对书中代码进行了校验,全书由湖北工业大学张明武教授修改定稿。

在本书的编写中,负责本书编辑出版工作的机械工业出版社华章公司策划编辑、教材部副主任朱劭老师和余洁编辑付出了大量辛勤的劳动。武汉大学何炎祥教授在百忙之中认真审阅了全书,提出了许多宝贵和中肯的意见。在此,谨向每一位关心和支持本书编写工作的各位朋友、老师表示衷心的感谢!

由于作者的知识 and 写作水平有限,书中难免有错误和不足之处,恳请各位专家、读者批评指正。作者 E-mail:nancy78733@126.com。

沈 华

2015 年 4 月于武汉南湖

第 1 章		第 1 章 绪论
第 2 章		第 2 章 基 础
第 3 章	3-1 ~ 3-8	第 3 章 基 础
第 4 章	4-1 ~ 4-2	第 4 章 基 础
第 5 章	5-1 ~ 5-5	第 5 章 基 础

教学建议

教学内容	教学要点及教学要求	课时安排	
		计算机专业	非计算机专业
第1章 数据结构	<ul style="list-style-type: none"> 熟悉并掌握数据、数据元素、数据项的概念 熟悉并掌握数据的逻辑结构、数据的物理结构、数据运算的概念,以及常见的几种数据的逻辑结构和物理结构,并充分理解数据结构的含义 熟悉并掌握数据类型、抽象数据类型的概念 	2	1~2 (选讲)
第2章 算 法	<ul style="list-style-type: none"> 熟悉并掌握算法的概念和算法的五个特性,以及算法和程序之间的区别和联系 充分理解算法和数据结构之间的关系 了解描述算法的方法 掌握一般的算法时间复杂度和空间复杂度的分析方法,能够对给定算法进行时间和空间的复杂度分析 	2	1~2 (选讲)
第3章 线 性 表	<ul style="list-style-type: none"> 熟悉并掌握抽象数据类型——线性表 熟悉并掌握线性表的几种存储结构和在不同存储结构上线性表基本运算的实现过程,重点掌握的存储结构是顺序表和单链表 能够对适合应用线性表的实际问题进行线性表抽象,并能够根据应用环境和应用要求为其选择合适的存储结构,同时能够给出实现算法 	8~10	6~10 (选讲)
第4章 栈	<ul style="list-style-type: none"> 熟悉并掌握抽象数据类型——栈,重点掌握栈与线性表之间的关系、栈的结构特性 熟悉并掌握栈的存储结构——顺序栈和链栈,以及在不同存储结构上栈的基本运算的实现过程 了解两个方向生成的栈 能够对适合应用栈的实际问题进行栈抽象,并能够根据应用环境和应用要求为其选择合适的存储结构,同时能够给出实现算法 	2~4	2 (选讲)
第5章 队 列	<ul style="list-style-type: none"> 熟悉并掌握抽象数据类型——队列,重点掌握队列与线性表之间的关系、队列的结构特性、队列和栈的共性与个性 熟悉并掌握队列的存储结构——链队列、顺序队列和循环队列,以及在不同存储结构上队列的基本运算的实现过程,重点掌握循环队列的产生、实现、判空条件和判满条件 了解双端队列 能够对适合应用队列的实际问题进行栈抽象,并能够根据应用环境和应用要求为其选择合适的存储结构,同时能够给出实现算法 	2~4	2 (选讲)

(续)

教学内容	教学要点及教学要求	课时安排	
		计算机专业	非计算机专业
第6章 串	<ul style="list-style-type: none"> 熟悉并掌握抽象数据类型——串, 重点掌握串与线性表之间的关系、空串与空格串的区别、串相等的概念、模式匹配的概念、串运算的特点 熟悉并掌握串的存储结构——顺序串、串的堆结构、串的块链结构, 以及在不同存储结构上队列的基本运算的实现过程, 重点掌握串的堆结构的存储思想 了解表示顺序串串长的显式方式和隐式方式 掌握朴素模式匹配算法和 KMP 算法 	4	2 (选讲)
第7章 数组及广义表	<ul style="list-style-type: none"> 熟悉并掌握抽象数据类型——数组, 重点掌握数组与线性表之间的关系、数组维数的概念, 明确一维数组与向量、二维数组与矩阵之间的关系 了解将多维数组映射到一维存储空间的方法, 重点掌握二维数组映射到一维存储空间的方法, 能够将二维数组元素的二维地址转换为二维存储地址, 深刻理解数组的“随机访问性” 了解特殊矩阵(包括特殊形状矩阵和稀疏矩阵)的概念以及对特殊矩阵进行压缩的意义和压缩的目标 熟悉并掌握几种常见特殊形状矩阵的压缩存储方法, 并且能够通过矩阵元素的行列下标得到其在压缩向量中的下标 熟悉并掌握稀疏矩阵的压缩存储方法, 重点掌握稀疏矩阵的三元组顺序表压缩存储方法, 以及基于三元组顺序表实现的矩阵的转置运算的算法 熟悉并掌握广义表的概念, 广义表与线性表之间的关系、广义表的性质、广义表的基本运算 了解广义表的存储结构 	6	4 (选讲)
第8章 树与森林	<ul style="list-style-type: none"> 熟悉并掌握(根)树的概念、相关术语, 掌握(根)树的几种存储结构以及每种存储结构的优缺点 熟悉并掌握树的创建方法 掌握树的先序、后序和层次遍历算法 了解树的相关应用 熟悉并掌握森林的概念, 森林的先序、中序、后序、层次遍历算法 	4	2~4 (选讲)
第9章 二叉树	<ul style="list-style-type: none"> 熟悉并掌握二叉树的概念, 深刻理解二叉树和树的区别, 熟悉并能灵活应用二叉树的五个性质, 掌握五个性质的证明思路, 掌握满二叉树和完全二叉树的概念、特点以及它们之间的关系 熟悉并掌握二叉树的几种存储结构, 以及每种存储结构的优缺点 熟悉并掌握二叉树的创建方法 熟悉并掌握二叉树的先序、中序、后序遍历的递归和非递归算法以及层次遍历的实现算法, 并能够灵活运用二叉树的遍历算法解决实际问题 	8	4~8 (选讲)

(续)

教学内容	教学要点及教学要求	课时安排	
		计算机专业	非计算机专业
第9章 二叉树	<ul style="list-style-type: none"> 掌握线索、线索二叉树、二叉树的线索化的概念和给二叉树加线索的方法,了解对线索二叉树进行遍历的方法以及线索在遍历过程中的作用 掌握哈夫曼树的概念、生成方法、特点以及应用(哈夫曼编码),掌握二叉排序树的概念、特点、生成(插入运算)和删除运算,了解平衡二叉树的概念和各种失去平衡情况下的调整规则 了解树、森林和二叉树的相互转换过程和树转换得到的二叉树的特点,并深刻体会二叉树的二叉链表和树的左孩子右兄弟表示法之间的联系 了解树和森林的遍历与对应二叉树的先序、中序遍历之间的关系 	8	4~8 (选讲)
第10章 图	<ul style="list-style-type: none"> 熟悉并掌握图的概念、相关术语,掌握图的两种常用存储结构和每种存储结构的特点以及图的创建算法 熟悉并掌握图的深度优先搜索遍历和广度优先搜索遍历,并能够灵活运用图的两种遍历算法解决实际问题 熟悉并掌握连通图的生成树、连通网的最小生成树的概念,掌握最小生成树的两种生成算法 了解什么是图的最短路径问题,重点掌握求解网络单源最短路径的算法和求解网络每对顶点间最短路径的算法 掌握有向无环图的两种主要应用——AOV网的拓扑排序和AOE网的关键路径,并深刻理解它们对实际工程和应用问题的指导意义 	8	4~8 (选讲)
第11章 查找	<ul style="list-style-type: none"> 熟悉并掌握查找的概念、分类以及其他相关概念,明确静态查找和动态查找的区别 重点掌握各种查找法和它们的性能评价,重点掌握二分查找、散列查找、二叉排序树(在第9章给出),深刻了解散列查找法与其他查找方法之间的区别 	4	4 (选讲)
第12章 排序	<ul style="list-style-type: none"> 熟悉并掌握排序的概念、分类以及其他相关概念,明确内部排序和外部排序的区别 重点掌握各种内排序法和它们的性能评价及稳定性,重点掌握直接插入排序法、冒泡排序法、快速排序法、直接选择排序法、堆排序法、归并排序法、基数排序法 了解磁盘的组成和工作原理 了解几种常用的外排序方法——多路平衡归并、置换选择排序、最佳归并树 	6~8	4~6 (选讲)
教学总学时建议		56~64	36~54

说明:

① 计算机专业本科教学使用本教材时,建议课堂授课学时数为56~64(包含习题课、课堂讨论等必要的课堂教学环节,实验另行安排学时),不同学校可以根据各自的教学要求和计划学时数酌情对教材内容进行取舍。

② 非计算机专业的师生使用本教材时可适当降低教学要求。若授课学时数少于54,建议主要学习几种常见的

数据结构——线性表、栈、队列、串、数组、树和图，有关算法分析、查找、排序的内容可以适当简化。

课堂教学建议：

作者编写本书的目的在于帮助读者理解和回答下面四个问题：什么是数据结构？有哪些数据结构？为什么需要数据结构？如何应用数据结构？全书第1~12章的内容均围绕上述四个问题进行展开，这些内容对于掌握数据结构的含义、几种常见的数据结构、灵活应用数据结构均是至关重要的。线性表是线性部分的基础，我们可以将栈、队列、串、数组看作线性表的某种应用。同时，能否充分掌握树和图的各种存储结构，能否基于存储结构实现各种树和图的应用，线性表掌握的好坏也起着关键性作用。因此建议将第3章作为重中之重的内容进行细致、全面、深入地讲解。非线性结构中的很多内容可以用来解决许多实际应用问题，如哈夫曼编码算法、二叉排序树、完全二叉树、深度优先搜索算法、广度优先搜索算法、最小生成树、最短路径算法等等。因此建议教师花费足够的学时讲解第9章和第10章。

实验教学建议：

- ① 实验一：线性表的应用。
- ② 实验二：线性结构部分的综合性实验。
- ③ 实验三：二叉树的创建以及二叉树遍历算法的应用。
- ④ 实验四：图的创建以及图遍历算法的应用。
- ⑤ 实验五：查找部分的综合性实验。
- ⑥ 实验六：内部排序部分的综合性实验。

目 录

序	2.4.1 性能度量的方法	18
前言	2.4.2 生成测试数据	19
教学建议	2.5 知识点小结	19
	习题	19
第一部分 概论部分		
第1章 数据结构	第二部分 线性部分	
1.1 什么是数据	第3章 线性表	22
1.2 什么是数据结构	3.1 线性表抽象数据类型	22
1.2.1 数据的逻辑结构	3.1.1 线性表的逻辑结构	22
1.2.2 数据的存储结构	3.1.2 线性表的基本运算	22
1.2.3 数据的运算	3.1.3 线性表的 ADT 描述	23
1.3 什么是数据类型	3.2 线性表的应用——两个一元多项式 相加	24
1.4 什么是抽象数据类型	3.2.1 问题描述与分析	24
1.5 知识点小结	3.2.2 问题求解	25
习题	3.3 线性表的实现	26
第2章 算法	3.3.1 顺序表	26
2.1 什么是算法	3.3.2 单链表	41
2.2 算法的描述	3.3.3 静态单链表	52
2.3 算法的性能分析	3.3.4 一元多项式相加问题的求解 实现	56
2.3.1 时间复杂度	3.4 线性表的其他实现及应用场景 分析	69
2.3.2 渐近符号	3.4.1 双(向)链表	69
2.3.3 空间复杂度	3.4.2 循环单(向)链表	71
2.3.4 复杂度分析举例	3.4.3 循环双(向)链表	74
2.4 算法的性能度量		

3.5 知识点小结	75	5.4 双端队列及队列应用场景举例	110
习题	75	5.4.1 双端队列	110
第4章 栈	77	5.4.2 队列应用场景举例	111
4.1 栈抽象数据类型	77	5.5 知识点小结	111
4.1.1 栈的逻辑结构	77	习题	112
4.1.2 栈的基本运算	78	第6章 串	113
4.1.3 栈的 ADT 描述	78	6.1 串抽象数据类型	113
4.2 栈的应用——表达式求解	79	6.1.1 串的逻辑结构	113
4.2.1 问题描述与分析	79	6.1.2 串的基本运算	113
4.2.2 问题求解	79	6.1.3 串的 ADT 描述	114
4.3 栈的实现	85	6.2 串的实现	114
4.3.1 顺序栈	85	6.2.1 串的顺序存储表示	114
4.3.2 链栈	88	6.2.2 串的堆分配存储表示	118
4.3.3 在表达式求解问题上的性能 分析与比较	91	6.2.3 串的块链存储表示	120
4.4 顺序栈的一种有趣实现——两个 方向生长的栈	91	6.3 串的模式匹配	124
4.5 栈与递归的天然联系	92	6.3.1 朴素的模式匹配算法	124
4.6 知识点小结	93	6.3.2 KMP 算法	128
习题	93	6.4 知识点小结	133
第5章 队列	95	习题	133
5.1 队列抽象数据类型	95	第7章 数组及广义表	134
5.1.1 队列的逻辑结构	95	7.1 数组的类型定义	134
5.1.2 队列的基本运算	95	7.1.1 数组的定义	134
5.1.3 队列的 ADT 描述	96	7.1.2 数组的性质	134
5.2 队列的应用——模拟舞伴配对 问题	96	7.1.3 数组的基本运算	134
5.2.1 问题描述与分析	96	7.2 多维数组的线性存储方法	135
5.2.2 问题求解	97	7.3 特殊矩阵的压缩存储	138
5.3 队列的实现	97	7.3.1 特殊形状矩阵的压缩存储	138
5.3.1 顺序队列	97	7.3.2 随机稀疏矩阵的压缩存储及其 运算	142
5.3.2 循环队列	101	7.4 广义表	154
5.3.3 链队列	107	7.4.1 广义表的基本概念	154
		7.4.2 广义表的性质	155
		7.4.3 广义表的基本运算	155

7.4.4 广义表的存储结构	156	9.2.1 需要解决的关键问题	194
7.5 知识点小结	159	9.2.2 关键问题的求解思路	195
习题	159	9.2.3 二叉树的存储结构	195
第三部分 非线性部分			
第8章 树与森林	162	9.2.4 方案的比较分析	203
8.1 认识树	162	9.3 二叉树的创建	203
8.1.1 (根)树的定义	162	9.3.1 问题描述与分析	203
8.1.2 基本术语	163	9.3.2 问题求解	203
8.1.3 树的基本运算	164	9.4 二叉树的遍历	204
8.2 树的实现	167	9.4.1 问题描述与分析	204
8.2.1 需要解决的关键问题	167	9.4.2 问题求解	208
8.2.2 关键问题的求解思路	167	9.4.3 二叉树遍历应用举例	211
8.2.3 树的存储结构	167	9.5 线索二叉树	215
8.2.4 存储方案的比较分析	178	9.5.1 线索二叉树的应用需求	215
8.3 树的创建	178	9.5.2 二叉树的线索化	217
8.3.1 问题描述与分析	178	9.5.3 线索二叉树上的运算	219
8.3.2 问题求解	179	9.6 二叉树的应用	227
8.4 树的遍历	180	9.6.1 哈夫曼树及其应用	227
8.4.1 问题描述与分析	180	9.6.2 二叉排序树及其应用	233
8.4.2 问题求解	180	9.6.3 平衡二叉树	236
8.5 树的应用	181	9.7 树、森林与二叉树的关系	241
8.5.1 并查集	181	9.7.1 树、森林与二叉树的相互 转换	241
8.5.2 等价类	182	9.7.2 树、森林与二叉树在遍历运算 上的关系	244
8.5.3 决策树	184	9.8 知识点小结	244
8.6 森林	184	习题	245
8.7 知识点小结	185	第10章 图	246
习题	185	10.1 认识图	246
第9章 二叉树	186	10.1.1 图的定义	246
9.1 认识二叉树	186	10.1.2 基本术语	246
9.1.1 二叉树的定义	186	10.1.3 图的基本运算	252
9.1.2 二叉树的基本运算	187	10.2 图的实现	253
9.1.3 二叉树的性质	189	10.2.1 需要解决的关键问题	253
9.2 二叉树的实现	194		

10.2.2 关键问题的求解思路	253	11.3 动态查找	304
10.2.3 图的存储结构	254	11.4 散列技术	312
10.2.4 存储方案的比较分析	261	11.4.1 散列表的概念	312
10.3 图的创建	262	11.4.2 散列函数的构造方法	312
10.3.1 问题描述与分析	262	11.4.3 处理冲突的方法	312
10.3.2 问题求解	262	11.4.4 散列表的查找	315
10.4 图的遍历	264	11.4.5 散列表的应用	317
10.4.1 问题描述与分析	264	11.5 知识点小结	318
10.4.2 深度优先搜索遍历	265	习题	318
10.4.3 广度优先搜索遍历	272	第 12 章 排序	319
10.4.4 图遍历的应用	277	12.1 排序的基本概念	319
10.5 生成树	277	12.2 插入排序	321
10.5.1 连通图的生成树	277	12.2.1 直接插入排序	321
10.5.2 连通网的最小生成树	278	12.2.2 希尔排序	324
10.6 最短路径	280	12.3 交换排序	325
10.6.1 单源最短路径	281	12.3.1 冒泡排序	325
10.6.2 每对顶点间的最短路径	284	12.3.2 快速排序	327
10.6.3 最短路径应用举例	288	12.4 选择排序	330
10.7 有向无环图及其应用	288	12.4.1 直接选择排序	330
10.7.1 AOV 网与拓扑排序	288	12.4.2 树形选择排序	332
10.7.2 AOE 网与关键路径	290	12.4.3 堆排序	333
10.8 知识点小结	293	12.5 归并排序	338
习题	293	12.5.1 (内部) 归并排序	338
		12.5.2 外部归并排序	339
		12.6 分配排序	339
		12.6.1 箱排序	339
		12.6.2 基数排序	339
		12.7 各种(内部)排序方法的比较	341
		12.8 知识点小结	342
		习题	342
		参考文献	344
第四部分 重要运算部分			
第 11 章 查找	296		
11.1 查找的基本概念	296		
11.2 静态查找	297		
11.2.1 顺序查找	297		
11.2.2 二分查找	299		
11.2.3 分块查找	303		

第一部分 概论部分

有用就会有需求，有了需求就会有动力，因此在开始本课程学习之前，我们必须弄清楚下面几个问题，这将有助于端正学习态度，明确学习目的并找到一种有效的学习方法。

第一个问题：我们为什么需要学习数据结构？

计算机领域分两个大方向：一个是硬件方向，另一个是软件方向。硬件方向又包括两个大的方面，一个是硬件器件技术的发展，另一个是如何在现有硬件条件下设计出高性能的计算机系统。软件是程序、数据以及文件的集合，它大致可分为系统软件和应用软件两大类。系统软件追求的是如何在方便用户使用的的前提下将计算机系统的性能发挥到极致。应用软件则是直接面对终端用户，它寻求的是更好的服务质量和更好的用户体验。我们将从应用软件开发者的视角去探寻数据结构的重要地位。

实际问题往往很复杂，当我们去求解实际问题的时候，一般是对实际问题的模型进行求解。所谓模型就是对实际问题的一个简化，是反映问题本质的数据集合以及数据之间关系的集合。对模型的求解则是对给定的输入找到对数据的一系列处理步骤，使得能够得到预期的输出。为了让计算机实现我们对模型的求解思路，则必须将模型映射到存储器中，这样计算机才能根据人的意图对操作对象（即数据）进行处理。

通过上面的分析可知，计算机的各种应用实际上都是对数据的处理，因此对数据以及它们之间关系的分析、表示是不可或缺的。

什么是结构？我们来看下面的例子，一个家族成员之间的双亲与孩子的关系构成了一个树形的家谱图；城市之间的互通关系构成了一张网状的交通图；操作系统各模块之间的单向依赖关系（即第 $i+1$ 层中的模块可以调用第 i 层中的模块，但第 i 层中的模块不能调用第 $i+1$ 层中的模块）构成了一个层次结构；长、宽、高之间的关系构成了三维空间结构；自然数集上的“ $<$ ”关系构成了一个有向无环图；人与人之间的不同社会关系构成了不同的社会结构。通过这些例子我们可以发现，结构是关系的一种表现形式，可以说结构即关系。数据结构就是数据以及数据之间的关系，它对更好地使用计算机来解决实际问题至关重要。

第二个问题：我们应该从这门课程学到什么？如何学？

我们将会从这门课程中学习几种典型的数据结构（这些数据结构基本涵盖了现实中的绝大多数关系类型）以及两种在实际问题中经常要涉及的运算（查找和排序）的各种实现算法。

我们要能清晰地知道每种数据结构描述的是数据之间怎样的一种关系，有哪些典型应用，它们在存储器中有哪几种存储表示，这些存储表示方式各有什么优缺点，不同数据结构之间有哪些联系和区别，查找和排序有哪几种不同的实现算法，这些算法在时间开销和空间开销方面的比较。

简单地说，我们应该从这门课程中学习各种相关的计算思维：描述和表示数据及数据之间关系的方法、将它们映射到存储器中的方法、根据数据的逻辑结构设计算法、根据数据的存储结构实现算法、根据实际应用选择合适的存储映射结构等等。

学习的目的是应用，将所学知识转化为自身的能力。因此，在学习数据结构的过程中，我

们要特别注重实践环节，在计算机上验证书中给出的各种数据结构的实现和应用算法，在此基础上应该尝试着用所学内容去解决实际问题。首先通过对问题的观察理解，分析出问题求解的对象、涉及的主要操作，然后通过联想寻找该问题与我们已有知识之间的联系，选用适当的数据结构描述问题的处理对象，最终达到解决问题的目的。

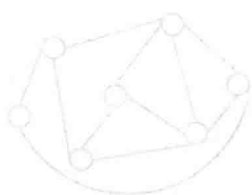
本部分下面的内容主要是帮助读者认识本课程的两大主角——数据结构和算法。

一个问题可以用不同的数据结构来描述，针对不同的结构它有多种不同的处理方法，有些方法效率高，有些则不然。也就是说，我们需要针对问题找到合适的数据结构，这样才能产生求解问题的高效方法。可见，数据结构和问题的处理方法（即算法）是密不可分的。瑞士著名科学家 N. Wirth 用下述著名公式生动描述了数据结构与算法之间的这种密切关系：

$$\text{程序} = \text{数据结构} + \text{算法}$$

下面是对这种密切关系的详细解读。

计算机求解应用问题有两个视图：抽象视图和实现视图。抽象视图关注的是数据之间已存在的某种关系，与机器无关，它体现出的结构称为数据的逻辑结构。算法设计者在抽象视图层面上设计抽象算法。实现视图主要解决如何让计算机“看见”数据以及数据之间的逻辑关系，以便计算机能够根据程序设计者设计的程序处理“看见”的数据。计算机的可视区是内存（当然还有高速缓存和寄存器），因此实现视图需要将数据和它们之间的逻辑关系存储到内存中。计算机在存储数据之间的关系时，数据对应的存储映像之间也会呈现出一种（新）结构，这种结构称为数据的存储结构。例如，线性表是抽象视图中的对象，元素之间的线性关系（逻辑关系）存储在存储器中可能仍然呈现出一种线性结构（顺序表），也可能呈现出一种非线性结构（链表）。程序设计者在实现视图层面上将抽象算法转换成具体的可执行程序。



第1章 数据结构



图 1-1-1 图状结构

图 1-1-2 树形结构

图 1-1-3 集合结构

图 1-1-4 散点结构

通过前面的讲述，我们已经对数据结构有了一个初步的印象：它是数据之间关系的描述。我们需要进一步了解的即什么是数据？数据之间常见的关系有哪些？这些关系在存储器中如何表示？“数据结构”和我们通常所说的“数据类型”、“抽象数据类型”之间有什么区别和联系？这些问题将在本章中一一得到解答。

1.1 什么是数据

简单地说，数据（Data）是描述客观事物且能被计算机识别并加工处理的对象，它可以是数值、字符、声音、图像等。

数据元素（Data Element）是数据的基本单位，是数据集合中的个体。在计算机程序中通常作为一个整体来考虑和处理。数据元素在不同的数据结构中有不同的称谓，如在树结构中称为结点，在图结构中称为顶点，在数据库表中称为记录等等。它可以由一个或多个数据项组成。

数据项（Data Item）是数据的最小单位，是不可再分的。因此仅由一个数据项构成的数据元素称为原子元素，由多个数据项构成的数据元素称为结构元素。

数据对象（Data Object）是性质相同的数据元素的集合，它是数据的一个子集。

例如，某班级的成绩表，就是一个数据对象，其中某个同学的成绩（对应于表中的一行）就是一个数据元素，构成该同学成绩的各个字段就相当于数据项。再如，整数集是一个数据对象，其中的每个整数是一个数据元素且为原子元素。

1.2 什么是数据结构

结构即关系，**数据结构（Data Structure）**是相互之间存在一种或多种特定关系的数据元素的集合。数据结构是一个二元组，记为： $data_structure = (D, S)$ ，其中 D 为数据元素的集合， S 是 D 上关系的集合。

通过前面的介绍可知，讨论数据结构的目的是，能够在抽象视图和实现视图中设计和实现高效的算法，使得我们能够高效地使用数据解决应用问题。因此，数据结构应该包括数据元素的逻辑结构、存储结构和相适应的运算三个方面的内容。

换句话说，在计算机中，数据结构是以高效使用数据为目标的存储和组织数据的一种特定方式。一种数据结构就是组织和存储数据的一种特殊格式。常见的数据结构有线性表、栈、队列、树、图等。

1.2.1 数据的逻辑结构

数据的逻辑结构（Logical Structure of Data）是指数据之间的逻辑关系。它与计算机无关，它可以作为从具体问题中抽象出来的数据模型。数据的逻辑结构通常有以下四种基本的结构类型：集合结构、线性结构、树形结构和网状结构（也称为图状结构），如图 1-1 所示。