清華大學 计算机系列教材

殷人昆 编著

数据结构（C语言版）（第2版）

清华大学出版社

清华大学 计算机系列教材

殷人昆 编著

数据结构（C语言版）（第2版）

清华大学出版社
北京

内 容 简 介

本书是根据教育部《高等学校计算机科学与技术专业公共核心知识体系与课程》编写的数据结构主教材。全书共 8 章。第 1 章介绍数据结构的地位和主要知识点，数据结构和算法的基本概念和算法分析的简单方法，以及 C 语言编程的要点。第 2～8 章分别介绍了线性表、栈和队列及其应用、多维数组、特殊矩阵、稀疏矩阵、字符串和广义表、树与二叉树、图、查找、排序，并做了适当延伸。作者在讨论每一个知识单元时，结合 30 多年教学的经验和考试辅导的体会，合理安排教材内容，力求透彻、全面，对学生读书容易忽略的地方和隐藏在书中所讨论问题后面的东西都有适当的提示。

本书的编写得到清华大学 2015 年精品教材建设项目的资助。本书既可作为高等学校计算机科学与技术专业和软件工程专业本科生学习数据结构与算法课程的教材，也可以作为计算机专业考研的辅导教材或其他计算机或软件考试的复习教材，还可作为计算机或软件系统开发人员的参考资料。

图书在版编目（CIP）数据

数据结构：C 语言版 / 殷人昆编著. —2 版. —北京：清华大学出版社，2017
（清华大学计算机系列教材）
ISBN 978-7-302-45989-7

Ⅰ. ①数… Ⅱ. ①殷… 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. ①TP311.12 ②TP312.8

中国版本图书馆 CIP 数据核字（2016）第 312857 号

责任编辑：龙启铭　战晓雷
封面设计：常雪影
责任校对：梁　毅
责任印制：杨　艳

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 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　印　　张：26.25　　字　　数：638 千字
版　　次：2012 年 10 月第 1 版　2017 年 5 月第 2 版　　印　　次：2017 年 5 月第 1 次印刷
印　　数：1～2000
定　　价：49.50 元

产品编号：068323-01

第 2 版前言

本书第 2 版的初稿完成于 2015 年 12 月，作为另一本教材《数据结构精讲与习题详解》（第 2 版）的写作参照，相互融合，相互补充，首先完成了该书，再回过头来第二次修改本书，所以本书实际上是第 2 版的修订本。

自从 1978 年美籍华人冀中田第一次在中国讲授“数据结构”课开始，很多老师对课程的内容和讲授方法做了大量的研究，也可以说是在做中学，总结出许多好的经验，使得课程的教学比当时进步了很多。我本人在这门课程的教学中也积累了一些心得，非常希望与正在学习这门课程的同学们分享，这是我修改这本教材的初衷。

既然数据结构与算法相辅相成，密不可分，而算法就是解决问题的过程描述，那么，描述数据结构与算法的语言就应该是过程性的。早期用伪代码描述，实践证明不可持续，因为很多用伪代码描述的算法转换为使用某种编程语言编写的程序后，怎么也通不过。原因是很多人编程语言的实践能力太差，算法的实现细节太粗糙。所以我认为，使用某种过程性语言，如 C、C++等，对于学习和实现数据结构与算法是合适的。

由于数据结构课程学时的限制（多数学校为 48～64 学时），作为本科生的教材，包含的知识容量应有一定限度。知识点太少，学生在未来的学习和工作中可联想的思考空间狭窄，使解决问题的能力受限；知识点太多，必然沦为百科全书式的阅读，基础不牢靠，同样使得解决问题的能力受限。通过教学实践证明，本书的第 1 版在内容上取材是恰当的，范围上取材的深度和广度也是恰当的，但联想不够，某些算法的实现上偏向使用伪代码描述，造成部分读者在学习上和实践上的困惑。本书第 2 版修改部分包括：

（1）在结构上从第 1 版的 10 章改为 8 章，虽然章数压缩了，但叙述内容不减反增。增加的知识点大多作为“扩展阅读”出现，它们不作为考核内容，主要是拓展视野。

（2）各章的“想想看”改为“思考题”，目的是增加一些互动环节。这些思考题触及的都是可联想的内容，或者是对理解正文有用的知识“点拨”。所谓“学问”就是有“学”还要有“问”。正面的“问”有助于理解“应该做什么”，反面的“问”有助于理解“不该做什么”。

（3）书中所有使用 C 语言书写的算法，包括辅助教材《数据结构精讲与习题详解》（第 2 版）中的 800 道算法题，都重新使用 VC++ 6.0 编译程序调试过，有的还按照软件工程的要求做了边界值测试。因为书中算法的正确运行需要构建运行环境，所以对于书中所涉及的主要数据结构的存储表示，绝大多数都在第 2 版给出了结构定义、初始化或创建算法、输出算法等，这样可由读者自行搭建运行环境。

（4）第 3 章增加了多栈共享同一存储时的栈浮动技术、递归程序的非递归模拟方法、优先队列的内容；第 4 章增加了 w 对角矩阵的压缩存储、稀疏矩阵的链表存储、串的 BM 模式匹配算法的内容；第 5 章增加了等价类与并查集的内容；第 6 章增加了构造最小生成树的破圈法、Dijkstra 算法的内容；第 7 章增加了跳表、红黑树、伸展树、字典树的内容。此外对保留的内容有部分增删。教材现有内容基本覆盖大多数学校的教学大纲和考研大纲。

（5）附录增加了词汇索引，书中出现的重要概念都收录在索引中，大大方便了读者查阅相关的词汇。

各章所附习题不包括选择题，但精选了许多综合应用题，这些习题的参考解答请参看作者的另一本配套教材《数据结构精讲与习题详解》。

因为作者的水平有限，可能在某些方面有考虑不周的地方，书中难免存在疏漏或错误，诚恳希望读者提出宝贵意见。

作者的 E-mail 地址是 yinrk@tsinghua.edu.cn 或 yinrk@sohu.com。

作　者

2016 年 8 月于清华大学

目　录

第1章 绪 论

开发一个计算机系统，最基本的工作就是编程。没有程序，没有数据，再好的计算机硬件也是没有用的。譬如修建高速铁路，没有好的运行计划程序、调度程序、网络通信程序和机车控制程序，不可能让整个线路运转起来。因此，一个计算机系统离不开程序和数据。算法和数据结构是程序的核心，是支持问题解决所采取的数据组织方式。

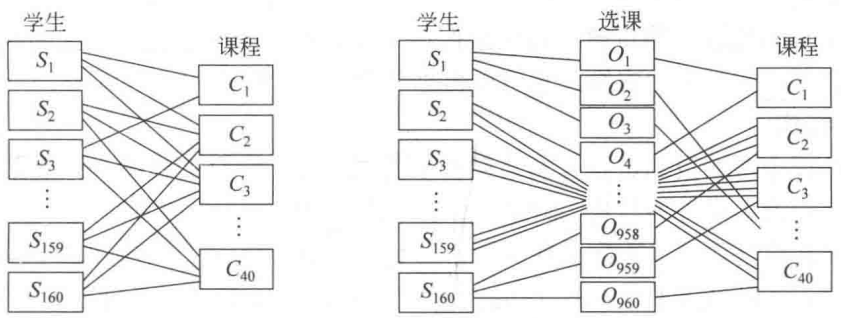
当前，算法和数据结构已成为计算机科学与技术学科和软件工程学科一门重要的专业基础课程，也是许多后续课程（如操作系统、计算机系统结构、计算机网络、编译原理、数据库、人工智能等）的先修课程。它不仅是计算机和软件工程专业的必修课，也是许多非计算机专业学生的选修课和系统开发人员的培训课程。

1.1 数据结构的概念及分类

1.1.1 为什么要学习数据结构

当今世界是一个互联网普及、信息大爆炸的世界，各种 APP（应用程序）已经为人们的生活提供了种种方便。不同领域的人们对各种 APP 的需求越来越广泛，然而，这都离不开计算机程序的开发人员。如何开发出性能优良、可靠性高的程序，是摆在计算机从业人员面前的首要问题，这些需求形成了一门基础性学科——程序设计方法学，而算法和数据结构正是程序设计方法学的核心。

【例 1-1】 开发大学选课系统。假设某学期为计算机系的 160 名本科生设置了 40 门可选课程，限定每位学生一个学期只能选 6 门课程。这样，学生和课程之间出现了多对多的关系，如图 1-1(a)所示。如果引入一个交互实体“选课”，学生和课程之间的关系就转化为两个一对多的关系，如图 1-1(b)所示。



(a) 一个多对多的关系　　(b) 两个一对多的关系

图 1-1　学生与课程之间的两种联系

图 1-1 给出了问题中出现数据之间的关系，这些关系包括一对一的、一对多的和多对多的关系。数据以及它们之间关系若选择得当，可以大大简化问题，设计出良好的解决方

案，降低问题解决的困难度，甚至可以大幅降低最终程序的运行时间或节省大量的存储空间。

事实上，问题的最终解决取决于程序，而程序的质量又取决于算法的选择和数据的组织方式。著名的瑞士科学家，图灵奖获得者 Niklaus Wirth（沃思）在其经典著作 *Algorithms＋Data Structures＝Programs* 中强调“程序的构成与数据结构是两个不可分割地联系在一起的问题。”他又引用 C. A. R. Hoare（霍尔）在 *Notes on Data Struncturing* 中的名言：“不了解施加于数据上的算法就无法决定如何构造数据，反之，算法的结构和选择却常常在很大程度上依赖于作为基础的数据结构。”从而给出一个权威性的定义：程序就是在数据的某些特定的表示方式和结构的基础上对抽象算法的具体表述。

因此，学习数据结构的意义在于编写高质量的程序。因为人们的直观概念总是数据先于算法——你总得先有对象才有施加于它的算法。

1.1.2 与数据结构相关的基本术语

1. 数据

我们在日常生活中会遇到各种信息，如用语言交流的思想，在战争中用于传递命令的旗语等。这些信息必须转换成数据才能在计算机中进行处理。因此，数据的定义是：数据是信息在计算机程序中的表示形式或编码形式，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识别和处理的符号的集合。

数据大致可分为两类：一类是数值性数据，包括整数、浮点数、复数、双精度数等，主要用于工程和科学计算，以及商业事务处理；另一类是非数值数据，主要包括字符和字符串，以及文字、图形、图像、语音等。

2. 数据元素

数据的基本单位是数据元素，它是计算机处理或访问的基本单位。例如，一个考生名册中的每个学生记录，一个字符串中的每一个字符，一个数组的每一个数组成分都是数据元素。不同场合下数据元素可以有别名，如元素、记录、结点、表项等。

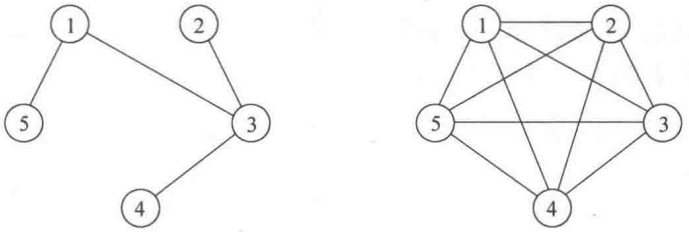
3. 数据项

数据元素可以是简单元素，如整数、浮点数、字符等；也可以是由多个数据项构成的复合元素。数据项又称为属性、字段、域。数据元素中的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，它可以再划分为物理、化学等更小的项。

4. 数据结构

在数据处理中所涉及的数据元素之间都不是孤立的，在它们之间存在着某种关系，这种数据元素之间的关系称为结构。例如，招生考试时把所有考生按考试成绩从高到低排队，所有考生记录都将处在一种有序的序列中；又如，在 n 个网站之间建立通信网络，要求以最小的代价将 n 个网站连通，如图 1-2(a)所示，这样，在所有网站之间形成一种树形关系；反之，要求当网络中任一网站出现故障时，整个网络仍然保持畅通，这样，在所有网站之间形成一种网状关系，如图 1-2(b)所示。

由此可以引出数据结构的定义：数据结构是由与特定问题相关的某一数据元素的集合和该集合中数据元素之间的关系组成的。



(a) 网站之间的树状关系　　(b) 网站之间的网状关系

图 1-2 n 个网站之间的连通关系

数据结构可分为静态数据结构（static data structure）和动态数据结构（dynamic data structure）。例如，数组是静态数据结构，它的元素个数和元素间的关系是不变的；链表和索引表是动态数据结构，它们的元素个数和元素间的关系将会因插入或删除而变化。

5. 数据对象

从狭义的观点把数据对象定义为具有一定关系的相同性质的数据元素的集合。从广义的观点把数据对象定义为一个由数据抽象和处理抽象构成的封装体，即数据对象的声明中不但要包含属性，还要包含可用的操作。

6. 数据类型

从程序设计角度来看，数据类型与数据结构的概念是相通的，主要用于刻画程序中操作对象的特性。数据类型明确地或隐含地规定了在程序执行期间变量、表达式或函数所有可能取值的范围，以及作用在这些取值上的操作。因此，数据类型是一个值的集合和定义在这个值集合上的一组操作的总称。例如，C 或 C++语言规定了一些内置的数据类型，如整数、浮点数、字符、指针、枚举量等，它们的表示、取值和操作如表 1-1 所示。

表 1-1 C/C++语言中的内置数据类型

表示	char	int	unsigned int	long int	unsigned long int	float	double
类型	字符型	整型	无符号整型	长整型	无符号长整型	浮点型	双精度型
位数	8b	16b	16b	32b	32b	32b	64b
取值	−127～127	−32 767～32 767	0～65 535	−2 147 483 647～2 147 483 647	0～4 294 967 295	6 位有效数字	10 位有效数字
操作		+，−，×，/，%	+，−，×，/，%	+，−，×，/，%	+，−，×，/，%	+，−，×，/	+，−，×，/

但仅有内置的数据类型还不能满足所有应用系统的需求，需要使用更复杂的数据类型，即构造型数据类型。内置数据类型在数据设计中亦称基本数据类型或原子类型，它的每个数据元素都是单一的无法再分割的整体。但构造数据类型可以由不同成分的内置数据类型或子结构类型按照一定的规则组成。例如，一个学生的学籍卡片是一个构造数据类型，除了包括姓名、性别、年龄等基本类型外，还包括如家庭成员等子结构类型。在编写 C 或 C++语言程序时人们可以自行定义为解决应用问题所必需的数据类型，它是确切地描述数据对象，正确地进行相关计算的有效工具。

【例 1-2】 数据表（Data List）的数据类型定义如程序 1-1 所示。

程序 1-1 数据表的构造型类型定义

```
#define maxSize 100                    //表空间的大小，可根据实际情况决定
typedef struct {
    int elem[maxSize];                 //存放表元素的向量
    int n;                             //当前的表长度
} DataList;
```

数据类型和数据结构又是什么关系呢？一般认为，数据结构是一种抽象的描述，数据元素的定义和元素间关系的定义舍弃了实际的物理的背景，是通用型的定义。数据类型是一种有实际问题要求背景的特定的定义，是数据结构的实例化。

数据类型分两种，一种是内置数据类型，如 int、float、double、char、string 等，是编程语言已经实现了的，程序员可直接在程序中使用的数据结构。而构造数据类型是程序员用编程语言描述的数据结构的存储映像。

数据对象和数据结构又是什么关系呢？从对象技术的意义上讲，数据对象不但包括了数据结构，还包括了数据结构的存储表示和施加于其上的运算。可以说，数据对象包含了数据结构所涉及的所有层面。

7. 抽象数据类型 ADT

在软件设计时，常常提到“抽象”和“信息隐蔽”。那么，什么是抽象呢？

抽象的本质就是抽取反映问题本质的东西，忽略非本质的细节。对于数据的抽象，可以用一个例子说明。在汇编语言中则给出了各种数据的自然表示，如 15.5、1.3E10、10 等，它们是二进制数据的抽象，编程人员在编写程序时可以直接使用它们，不必考虑实现的细节。到了高级语言，出现了整型、实型、字符型、双精度型等，它们是更高一级的数据抽象。为适应现代程序设计技术的发展，又出现了抽象数据类型。它可以进一步定义更高级的数据抽象，如各种表、队列、图，甚至窗口、管理器等。编程人员只需了解这种数据抽象包含哪些信息，有哪些可用的服务，即可在自己的程序中使用它。

抽象数据类型还有一种特性，即使用与实现分离，实现封装和信息隐蔽，就是说，抽象数据类型把数据的具体实现封装在数据类型之内，将其隐蔽起来。使用者在使用这种数据类型支持问题的解决时不必考虑类型中数据的具体组织和相关操作的编程细节，只要通过相关对外公开操作的调用（接口）来使用即可。这样做的好处是提高了复用程度，把可能的修改局部化。如果把数据类型内部的实现改掉，只要接口的调用形式不变，用户程序中所有调用此接口的地方一律可以不改变，提高了程序的可修改性和可移植性。

一般地，抽象数据类型由用户定义，是用以表示应用问题的数据模型。抽象数据类型不像 C 语言中的构造（struct）类型那样，把数据结构和相关操作分别定义，而是把数据成分和一组相关的操作封在一起。因此，抽象数据类型在 C++、Java 中可以用“类”直接描述，在 C 语言没有适当的机制描述，所以本书在以后的讨论中，不再涉及抽象数据类型。

在使用 C 语言编程时，要求先用 typedef struct…来定义数据的结构类型，再分别定义相关的操作。从教学观点来看，用 C 语言描述比较简洁，初学的学生容易上道。

思考题 系统开发中数据设计三视图为数据内容、数据结构和数据流。其中有关数据