



范例程序和
课后习题
源代码下载

丰富的图文详解
C#范例注释



吴灿铭 胡昭民 编著

图解数据结构 使用C#

丰富的图示描述复杂的
数据结构概念

完整的算法及范例程序
阐明数据结构的内涵

以C#语言实现数据结
构的重要理论

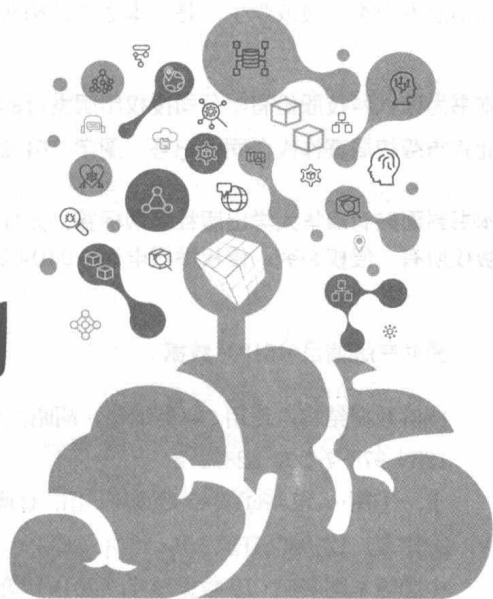
大量的课后习题供上机
练习和检验

清华大学出版社



图解数据结构 使用C#

吴灿铭 胡昭民 编著



清华大学出版社
北京

内 容 简 介

这是一本综合讲述数据结构及其算法的入门书，全书采用图文讲解的方式，力求读者易于学习和掌握。

全书从基本的数据结构概念开始讲起，包括数组结构、队列、堆栈、树形结构、排序、查找等；接着介绍常用的算法，包括分治法、递归法、贪心法、动态规划法、迭代法、枚举法、回溯法等，并为每个经典的算法都提供了 C# 程序设计语言编写的完整范例程序；最后在每章末尾都安排了大量的习题，这些题目包含各类考试的例题，希望读者能灵活地应用所学的各种知识。

本书图文并茂，叙述简洁、清晰，范例丰富，可操作性强，针对具有一定编程能力又想提高编程“深度”的非信息专业类人员或学生，是一本数据结构普及型的教科书或自学参考书。

本书为荣钦科技股份有限公司授权出版发行的中文简体字版本

北京市版权局著作权合同登记号 图字：01-2019-0484

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目 (CIP) 数据

图解数据结构：使用 C#/吴灿铭，胡昭民编著. —北京：清华大学出版社，2019

ISBN 978-7-302-52872-2

I. ①图… II. ①吴… ②胡… III. ①数据结构—图解②C 语言—程序设计

IV. ①TP311.12-64②TP312.8

中国版本图书馆 CIP 数据核字 (2019) 第 083047 号

责任编辑：夏毓彦

封面设计：王 翔

责任校对：闫秀华

责任印制：宋 林

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

印 装 者：清华大学印刷厂

经 销：全国新华书店

开 本：190mm×260mm

印 张：28.25

字 数：723 千字

版 次：2019 年 7 月第 1 版

印 次：2019 年 7 月第 1 次印刷

定 价：79.00 元

产品编号：081834-01

序

数据结构一直是计算机科学领域非常重要的基础课程,其除了是全国各大院校信息工程、信息管理、通信工程、应用数学、金融工程(计算金融)、计算机科学等信息类相关科系的必修科目外,近年来包括电机、电子,甚至一些商学院管理科系也将数据结构列入选修课程。同时,一些信息类相关科系的研究生入学考试、专业等级考试等,数据结构也被列入必考科目。由此可知,无论是从考试的角度还是研究信息科学理论知识的角度来看,数据结构确实是有志于从事信息类工作人员必须重视的一门基础课程。

但是,要学好数据结构的关键在于能否找到一本既易于阅读,又能将数据结构中各种重要理论知识及其算法进行详细诠释并举例示范的书籍。本书是一本讲述如何将数据结构概念以 C# 程序设计语言来实现的著作,为了方便读者学习,书中的算法尽量不以伪代码来说明,而是以 C# 程序设计语言来实现完整的范例程序,这样不仅可以避免片断学习造成的困扰,同时也方便老师的教学和对程序代码的解说。

本书的所有范例程序都是在 Visual Studio 2017 环境下进行编写、编译、调试与运行的,是一套多种程序设计语言的集成开发环境,其版本分为三种: Visual Studio Community 2017、Visual Studio Professional 2017 和 Visual Studio Enterprise 2017。其中 Visual Studio Community 2017 是一个免费版本,主要提供给初学者使用,本书就采用了这个版本。在本书最后的附录中包含了有关 Visual Studio Community 2017 这个集成开发环境下载、安装与设置的简介。另外,为了帮助读者在以 C#语言实现数据结构程序的过程中可以精准地使用各种程序命令的正确语法,以及提高程序的调试效率,我们在附录中也整理了实现数据结构必备的 C# 程序命令,并以摘要的方式帮助读者快速掌握其中的重点。

我想一本好的理论书籍除了内容的专业性外,更需要有清楚易懂的结构安排,在细细阅读本书之后,相信读者可以体会笔者的用心,也希望本书能帮助读者对这门基础学科有更加全面的认识。

编 者

2019 年 5 月

改编说明

“数据结构”不仅仅只是讲授数据的结构以及在计算机内如何存储和组织数据的方式，它背后真正蕴含的是与之息息相关的算法，精心选择的数据结构配合恰如其分的算法就意味着数据或信息在计算机内被高效率地存储和处理。算法是数据结构的灵魂，它既神秘又“好玩”，简而言之：数据结构 + 算法 = “聪明人在计算机上的游戏”。

为了方便老师教学或读者自学，作者在描述数据结构原理和算法时文字清晰且严谨，并为每个算法及其数据结构提供了演算的详细图解。另外，为了便于教学中让学生上机实践或者自学者上机“操练”，本书为每个经典的算法都提供了 C# 程序设计语言编写的完整范例程序，并且每个范例程序都经过了测试和调试。本书的所有范例程序都可以在 Visual Studio 2017 的所有版本上顺利运行，请扫描下面二维码，获得这些范例程序（包含完整源代码）。



学习本书需要有面向对象程序设计语言的基础，如果读者没有学习过任何面向对象的程序设计语言，那么建议读者还是先学习一下 C# 程序设计语言再来学习本书；如果读者已经掌握了 Java、C++、Python 等任何一种面向对象的程序设计语言，即便没有学习过 C# 语言，只需要找一本“C# 程序设计语言快速入门”方面的参考书快速浏览一下或者参考本书附录 A，就可以开始本书的学习。

资深架构师 赵军
2019 年 5 月

目 录

第 1 章 数据结构与算法	1
1.1 数据结构的定义	2
1.1.1 数据与信息	2
1.1.2 数据的特性	3
1.2 算法	4
1.2.1 到处都是算法	4
1.2.2 算法的定义	5
1.3 算法性能分析	7
1.3.1 Big-oh	8
1.3.2 Ω (omega)	11
1.3.3 θ (theta)	11
1.4 常见算法介绍	11
1.4.1 分治法	12
1.4.2 递归法	13
1.4.3 贪心法	17
1.4.4 动态规划法	17
1.4.5 迭代法	18
1.4.6 枚举法	20
1.5 程序设计简介	21
1.5.1 程序开发流程	22
1.5.2 结构化程序设计	22
1.5.3 面向对象程序设计	23
课后习题	26
第 2 章 数组结构	28
2.1 线性表简介	29
2.2 认识数组	30
2.2.1 一维数组	31
2.2.2 二维数组	33
2.2.3 三维数组	37
2.2.4 n 维数组	40

2.3 矩阵	41
2.3.1 矩阵相加	41
2.3.2 矩阵相乘	44
2.3.3 转置矩阵	47
2.3.4 稀疏矩阵	49
2.3.5 上三角形矩阵	52
2.3.6 下三角形矩阵	57
2.3.7 带状矩阵	62
2.4 数组与多项式	62
课后习题	65
第 3 章 链表	67
3.1 动态分配内存	68
3.2 单向链表	69
3.2.1 建立单向链表	70
3.2.2 单向链表节点的删除	75
3.2.3 单向链表插入新节点	79
3.2.4 单向链表的反转	85
3.2.5 单向链表的串接	89
3.2.6 多项式链表表示法	90
3.3 环形链表	97
3.3.1 环形链表新节点的插入	97
3.3.2 环形链表中节点的删除	98
3.3.3 环形链表的串接	101
3.3.4 疏矩阵链表表示法	106
3.4 双向链表	108
3.4.1 双向链表的定义	108
3.4.2 双向链表节点的插入	109
3.4.3 双向链表节点的删除	110
课后习题	114
第 4 章 堆栈	116
4.1 堆栈简介	117
4.1.1 用数组来实现堆栈	118
4.1.2 用链表来实现堆栈	124
4.2 堆栈的应用	128
4.2.1 汉诺塔问题	130

4.2.2 老鼠走迷宫	136
4.2.3 八皇后问题	142
4.3 算术表达式的求值法 (对应于表达式的表示法)	146
4.3.1 中序表示法求值	147
4.3.2 前序表示法求值	148
4.3.3 后序表示法求值	149
4.4 中序法转换为前序法	150
4.4.1 二叉树法	150
4.4.2 括号法	150
4.4.3 堆栈法	151
4.5 前序与后序表达式转换成中序表达式	157
4.5.1 括号法	158
4.5.2 堆栈法	159
课后习题	161
第 5 章 队列	163
5.1 认识队列	164
5.1.1 队列的基本操作	164
5.1.2 用数组实现队列	165
5.1.3 用链表实现队列	169
5.2 队列的应用	171
5.2.1 环形队列	172
5.2.2 双向队列	176
5.2.3 优先队列	179
课后习题	180
第 6 章 树	182
6.1 树的基本概念	183
6.2 二叉树简介	185
6.2.1 二叉树的定义	185
6.2.2 特殊二叉树简介	187
6.3 二叉树存储方式	188
6.3.1 一维数组表示法	188
6.3.2 链表表示法	191
6.4 二叉树遍历	193
6.4.1 中序遍历	194
6.4.2 后序遍历	194

6.4.3	前序遍历	195
6.4.4	二叉树遍历的实现	196
6.4.5	二叉运算树	200
6.5	二叉树的高级研究	206
6.5.1	二叉排序树	206
6.5.2	二叉查找树	212
6.5.3	线索二叉树	216
6.6	树的二叉树表示法	222
6.6.1	树转化为二叉树	222
6.6.2	树林转化为二叉树	224
6.6.3	树与森林的遍历	225
6.6.4	确定唯一二叉树	228
6.7	优化二叉查找树	230
6.7.1	扩充二叉树	230
6.7.2	霍夫曼树	231
6.8	平衡树	232
6.8.1	平衡树的定义	232
6.8.2	B 树	234
	课后习题	235
第 7 章	图	238
7.1	图论简介	239
7.1.1	图的定义	240
7.1.2	无向图	240
7.1.3	有向图	241
7.2	图的数据表示法	242
7.2.1	邻接矩阵法	242
7.2.2	邻接链表法	247
7.2.3	邻接复合链表法	251
7.2.4	索引表格法	253
7.3	图的遍历	254
7.3.1	深度优先法	254
7.3.2	广度优先查找法	258
7.4	生成树	262
7.5	最小生成树	264
7.5.1	Prim 算法	264
7.5.2	Kruskal 算法	265

7.6 图的最短路径.....	271
7.6.1 单点对全部顶点——Dijkstra 算法与 A*算法.....	272
7.6.2 两两顶点间的最短路径——Floyd 算法.....	279
7.7 AOV 网络与拓扑排序.....	284
7.8 AOE 网络.....	286
课后习题.....	288
第 8 章 排序.....	292
8.1 排序简介.....	294
8.1.1 排序的分类.....	294
8.1.2 排序算法分析.....	294
8.2 内部排序法.....	295
8.2.1 冒泡排序法.....	296
8.2.2 选择排序法.....	301
8.2.3 插入排序法.....	305
8.2.4 希尔排序法.....	307
8.2.5 合并排序法.....	311
8.2.6 快速排序法.....	312
8.2.7 堆积排序法.....	316
8.2.8 基数排序法.....	323
8.3 外部排序法.....	326
8.3.1 直接合并排序法.....	326
8.3.2 k 路合并法.....	337
8.3.3 多相合并法.....	337
课后习题.....	338
第 9 章 查找.....	342
9.1 常见的查找方法.....	343
9.1.1 顺序查找法.....	344
9.1.2 二分查找法.....	346
9.1.3 插值查找法.....	349
9.1.4 斐波拉契查找法.....	352
9.2 哈希查找法.....	355
9.3 常见的哈希法.....	356
9.3.1 除留余数法.....	356
9.3.2 平方取中法.....	357
9.3.3 折叠法.....	358

9.3.4 数字分析法	359
9.4 碰撞与溢出问题的处理	359
9.4.1 线性探测法	360
9.4.2 平方探测法	362
9.4.3 再哈希法	362
9.4.4 链表法	364
9.4.5 哈希法综合范例	367
课后习题	370
附录 A C#开发环境与指令摘要	373
A.1 Visual Studio Community 2017 软件下载与安装	374
A.2 创建第一个控制台项目	376
A.3 C# 语言重要指令的简介	378
A.3.1 注释指令	379
A.3.2 控制台应用程序输入/输出指令	379
A.3.3 变量与常数	380
A.3.4 数组的声明与使用	381
A.3.5 数组的排序	382
A.3.6 随机数的使用	383
A.3.7 数据类型转换	383
A.3.8 对象与类	384
A.3.9 静态类与静态字段	386
附录 B 习题答案	388

第1章

数据结构与算法

计算机 (Computer) 是一种具备了数据计算与信息处理功能的电子设备。它可以接受人类所设计的指令或程序设计语言, 经过运算处理后, 输出期待的结果。

对于有志于从事信息技术专业领域的人员来说, 数据结构 (Data Structure) 是一门与计算机硬件和软件息息相关的学科, 称得上是从计算机问世以来经久不衰的热门学科。这门学科研究的重点是在计算机程序设计领域, 即研究如何将计算机中相关数据或信息的组合, 以某种方式组织起来进行有效地加工和处理, 其中包含算法 (Algorithm)、数据存储的结构、排序、查找、树、图及哈希函数等。

1.1 数据结构的定义

我们可以将数据结构看成是在数据处理过程中的一种分析、组织数据的方法与逻辑, 它考虑到了数据之间的特性与相互关系。简单来说, 数据结构的定义就是一种程序设计优化的方法论, 它不仅讨论到存储的数据, 同时也考虑到彼此之间的关系与运算, 使之达到加快执行速度与减少内存占用空间的作用, 如图 1-1 所示。

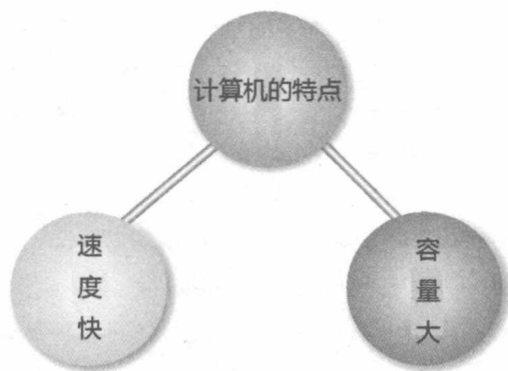


图 1-1

在现代社会中, 计算机与信息是息息相关的, 因为计算机有处理速度快与存储容量大两大特点, 所以在数据处理的角色上更为举足轻重。数据结构无疑就是数据进入计算机内处理的一套完整逻辑, 就像程序设计师必须选择一种数据结构来进行数据的添加、修改、删除、存储等操作。如果在选择数据结构时做了错误的决定, 那么程序执行的速度可能会变得非常低效; 如果选错了数据类型, 那么后果更是不堪设想。

因此, 当要求计算机为我们解决问题时, 必须以计算机所能接受的模式来确认问题, 而安排适当的算法处理数据, 这就是数据结构要讨论的重点。

1.1.1 数据与信息

谈到数据结构, 首先必须了解何谓数据 (Data) 与信息 (Information)。所谓数据 (Data), 指的就是一种未经处理的原始文字 (Word)、数字 (Number)、符号 (Symbol) 或图形 (Graph) 等, 它所表达出来的只是一种没有评估价值的基本元素或项目。例如, 姓名、课程表、通讯录等都皆可称为一种“数据” (Data)。

当数据经过处理（Process），例如以特定的方式整理、归纳甚至进行分析后，就成为“信息”（Information），而这样的处理过程就称为“数据处理”（Data Processing），如图 1-2 所示。



图 1-2

从严谨的角度来形容“数据处理”，就是用人力或机器设备对数据进行系统地整理，如记录、排序、合并、整合、计算、统计等，以使原始的数据符合需求，从而成为有用的信息。

大家可能会有疑问：“那么数据和信息的角色是否绝对一成不变呢？”。这倒也不一定，同一份文件可能在某种情况下为数据，而在另一种情况下则为信息。

1.1.2 数据的特性

通常按照计算机中所存储和使用的对象将数据分为两大类：一类为数值数据（Numeric Data），如 0, 1, 2, 3...9，即可用运算符（Operator）来进行运算的数据；另一类为字符数据（Alphanumeric Data），如 A, B, C...+, *等非数值数据（Non-Numeric Data）。如果按照数据在计算机程序设计语言中的存在层次来划分，则可以分为以下三种类型。

■ 基本数据类型（Primitive Data Type）

不能以其他类型来定义的数据类型，或者称为标量数据类型（Scalar Data Type），几乎所有的程序设计语言都会为标量数据类型提供一组基本数据类型，如 C#语言中的基本数据类型就包括了整数（int）、浮点（float）、字符（char）等。

■ 结构数据类型（Structured Data Type）

结构数据类型也称为虚拟数据类型（Virtual Data Type），是一种比基本数据类型更高一级的数据类型，如字符串（string）、数组（array）、指针（pointer）、列表（list）、文件（file）等。

■ 抽象数据类型（Abstract Data Type: ADT）

我们可以将一种数据类型看成是一种值的集合，以及在这些值上进行的运算及其代表的属性所组成的集合。“抽象数据类型”（Abstract Data Type, ADT）比结构数据类型更高级，是指一个数学模型及定义在此数学模型上的一组数学运算或操作。也就是说，ADT 在计算机中表示一种“信息隐藏”（Information Hiding）的程序设计思想及信息之间的某一种特定的关系模式。例如堆栈（Stack）就是一种典型的数据抽象类型，它具有后进先出（Last In, First Out）的数据操作方式。

1.2 算法

随着信息与网络科技的高速发展，在目前这个物联网（Internet of Things, IOT）与云运算（Cloud Computing）的时代，程序设计能力已经被看成是国力的象征，有条件的中小学校都将程序设计（或称为“编程”）列入学生信息课的学习内容，在大专院校里，程序设计已不再只是信息技术相关科系的“专利”了。程序设计已经是接受全民义务制教育的学生们应该具备的基本能力，只有将“创意”通过“设计过程”与计算机相结合，才能让新一代人才轻松应对这个快速变迁的云计算时代（图 1-3）。

没有最好的程序设计语言，只有是否适合的程序设计语言。程序设计语言本来就只是工具，从来都不是算法的重点。我们知道，一个程序能否快速而高效地完成预定的任务，算法才是其中的关键因素。所以，我们可以认为“数据结构加上算法等于可执行程序”，如图 1-4 所示。



图 1-3

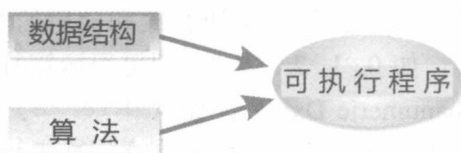


图 1-4



提示

“云”其实泛指“网络”，因为工程师在网络结构示意图中通常习惯用“云朵”图来代表不同的网络。云计算是指将网络中的运算能力提供出来作为一种服务，只要用户可以通过网络登录远程服务器进行操作，就能使用这种运算资源。

物联网（Internet of Things, IOT）是近年来信息产业中一个非常热门的话题，各种配备了传感器的物品，如 RFID、环境传感器、全球定位系统（GPS）等与因特网结合起来，并通过网络技术让各种实体对象、自动化设备彼此沟通与交换信息，也就是通过网络把所有东西都连接在一起。

1.2.1 到处都是算法

算法（Algorithm）是计算机科学中程序设计领域的核心理论之一，每个人每天都会用到一些算法。算法也是人类使用计算机解决问题的技巧之一，它不仅用于计算机领域，而且在数学、物理甚至是每天的生活中都应用广泛。在日常生活中就有许多工作可以使用算法来描述，例如员工的工作报告、宠物的饲养过程、厨师准备美食的食谱、学生的课程表等。如今我们几乎每天都在使用的各种搜索引擎也必须借助不断更新的算法来运行，如图 1-5 所示。



图 1-5

特别是在算法与大数据的结合下，这门学科演化出“千奇百怪”的应用，例如当我们拨打某个银行信用卡客户服务中心的电话时，很可能就先经过后台算法的过滤，帮我们找出一位最“合我们胃口”的客服人员来与我们交谈。在互联网时代，通过大数据的分析，网店还可以进一步了解产品购买和需求的人群，甚至一些知名 IT 企业在面试过程中也会测验面试人员对算法的了解程度（图 1-6）。



图 1-6



提示

大数据（又称为海量数据，big data），由 IBM 公司于 2010 年提出，是指在一定时效（Velocity）内进行大量（Volume）、多样性（Variety）、低价值密度（Value）、真实性（Veracity）数据的获得、分析、处理、保存等操作，主要特性包含：Volume（大量）、Velocity（时效性）、Variety（多样性）、Value（低价值密度）和 Veracity（真实性）。大数据解决了商业智能无法处理的非结构化与半结构化数据。

1.2.2 算法的定义

在韦氏辞典中算法定义为：“在有限步骤内解决数学问题的程序。”如果运用在计算机领域中，我们也可以把算法定义成：“为了解决某项工作或某个问题，所需要有限数量的机械性或重复性指令与计算步骤。”

算法必须符合的 5 个条件可参考表 1-1 和图 1-7 所示。

表 1-1 算法必须符合的 5 个条件

算法的特性	内容与说明
输入 (Input)	0 个或多个输入数据，这些输入必须有清楚的描述或定义
输出 (Output)	至少会有一个输出结果，不能没有输出结果
明确性 (Definiteness)	每一个指令或步骤必须是简洁明确的
有限性 (Finiteness)	在有限步骤后一定会结束，不会产生无限循环
有效性 (Effectiveness)	步骤清晰且可行，能让用户用纸笔计算而求出答案

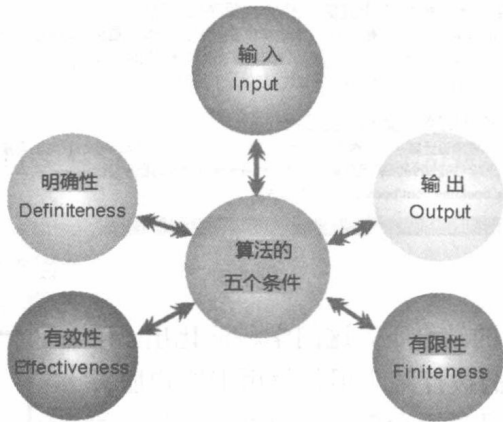


图 1-7

我们认识了算法的定义与条件后，接着要来思考：用什么方法来表达算法比较合适呢？其实算法的主要目的在于让人们了解所执行工作的流程与步骤，只要清楚地体现出算法的 5 个条件即可。常用的算法如下：

- 常用的算法一般可以用中文、英文、数字等文字来描述，即用语言来描述算法的具体步骤。例如，小华早上去上学并买早餐的简单文字算法，如图 1-8 所示。
- 伪语言 (Pseudo-Language) 是接近高级程序设计的语言，也是一种不能直接放进计算机中执行的语言。一般需要一种特定的预处理器 (Preprocessor)，或者要用人工编写转换成真正的计算机语言，经常使用的有 SPARKS、PASCAL-LIKE 等。以下是用 SPARKS 写成的链表反转的算法。



图 1-8

```
Procedure Invert(x)
  P←x;Q←Nil;
  WHILE P≠NIL do
    r←q;q←p;
    p←LINK(p);
```