用C实现数据结构程序设计

马春江 编著

清华大学出版社

用C实现数据结构程序设计

马春江 编著

清华大学出版社
北京

内容简介

本书的特色是在源码级别而不是算法级别上讨论数据结构，给出的程序构建能帮助学生掌握数据结构程序设计和提高综合运用数据结构的能力。全书共分15章，按照基础知识、理论知识和应用三部分来编写。第一部分包括数据结构的基本概念、C语言复习与归纳、递归思想与程序之一；第二部分包括线性数据结构、非线性数据结构；第三部分包括查找、排序等应用。

本书对于数据结构的综合运用进行了较为深入的讨论，在索引结构、广义表及文件结构等方面给出的程序源码将极大地提高学生对于数据结构编程的理解。

本书可作为高等院校理论与应用型本科层次计算机相关专业教材，也可作为高职高专层次各类学校的参考教材，还可作为计算机岗位培训和计算机爱好者的自学用书。

图书在版编目(CIP)数据

用C实现数据结构程序设计/马春江编著. --北京：清华大学出版社，2015
ISBN 978-7-302-38881-4

Ⅰ. ①用…　Ⅱ. ①马…　Ⅲ. ①C语言－程序设计 ②数据结构　Ⅳ. ①TP312 ②TP311.12

中国版本图书馆CIP数据核字(2015)第004728号

责任编辑：刘向威　王冰飞
封面设计：文　静
责任校对：时翠兰
责任印制：沈　露

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 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张**：23.5　　**字　数**：572千字
版　　次：2015年5月第1版　　**印　次**：2015年5月第1次印刷
印　　数：1～2000
定　　价：39.00元

产品编号：058747-01

前　言

任何希望掌握计算机程序设计的读者首先要了解由三门课程构成了程序设计理论底层的黄金三角形，分别是高级语言、数据结构、算法设计与分析。高级语言偏重语法的描述和细节程序设计等能力培养；数据结构重点讨论程序设计中如何分析、规划和存储实现相关的数据以及关系；算法设计则偏重解题思路的实现。大部分算法设计首先依赖于数据结构的构造，这将深远影响到程序的时间效率和空间效率，以及程序结构的合理性和程序阅读的简易性。计算机界著名人士尼克劳斯·沃思(Wirth N)提出的“算法＋数据结构＝程序”的观点正说明数据结构的重要性，即使在面向过程转向面向对象程序设计的今天，对象底层的函数实现依然能体现这句至理名言的准确性。

本人从事数据结构教学已有近三十年，多年教学实践中深刻体会到数据结构是计算机专业的一门很难的课程，学生普遍反映过于抽象和难于编程实现。这反映了几个现实问题：第一，高级语言课程所教授的内容离数据结构编程需求有一定的距离，高级语言教程通常更多讨论的是数值计算方面的程序设计范例，对于离散结构的讨论相对偏少；第二，算法设计是数据结构的后续课程，很多设计思想在数据结构课程中暂时无法起到引导作用；第三，从数据结构本身来看，教材结构或书写方式有时成了学习过程的阻碍。部分数据结构教材偏重理论，全部用算法表述数据结构相关程序设计思想，导致学生可以模仿编程的范例不够，实际效果不大理想。部分教材由于源于翻译国外教材的原因，术语和描述生涩难懂。部分教材重点难点不突出，整体结构不完整，增加了学生的学习困难。

2012年我在清华大学出版社主编出版了《数据结构与程序构建》(C++版)，受到了来自全国各地大学生或程序开发工作者的欢迎。他们告诉我，由于我的教材他们能够以最快的速度学习到数据结构的程序设计，效果很理想，并且希望我能推出更多平台的数据结构源码设计。受到这些热情读者和编辑的鼓励，我开始着手编写数据结构的C版程序设计。经过了一段非常值得怀念的日子，我的努力变成了现在这本教材。希望它能为那些只学习过C语言的学生快速掌握数据结构编程做出自己的贡献。

学习数据结构是一种“痛并快乐着”的过程，在学习它的时候大部分学生都会感到过于抽象和高深，但是如果能用程序具体实现，就会体会到激动的成就感，会多次被计算机科学家的奇思妙想所震撼，体会到程序设计的引人入胜，体会到整个过程是充满挑战和乐趣的。

本书最大的特色就是全面给出数据结构的相关程序构建源码，使得学生有一个可以研究、探讨、模仿、提高的平台。本书提供的程序构建范例都具有实用性和趣味性，覆盖了多种程序设计方法和界面设计风格，供学生研究使用。

全书体系结构完整，注重原理与实践结合，重点和难点突出，为学生搭建了一个很全面的学习研究平台。其目标是学生易于学习、教师易于组织教学。鉴于部分学生C语言的编

程能力不能满足数据结构多功能同时提供、程序反复运行、处理意外情况等要求，本书全面提供了各种程序界面的细节设计。数据输入方面提供了键盘输入、内部预置、随机产生、文件读入等多种方式供读者模仿学习。递归思想在数据结构的讨论中大量出现，本书单独一章先行讨论，给学生提供了一个良好的基础。排序在完成线性表之后先讲基本部分，体现了线性表的实际使用，而进阶部分涉及更复杂的数据结构，放在大部分数据结构学习完成之后讲解。另外，为了较为全面地体现数据结构的整体关系，本书专门讨论了广义表的实现，这部分构成了线性结构和非线性结构的桥梁，供学生了解掌握。

全书共分15章，按照数据结构学习的基础知识、理论知识和应用三大部分来编写。第一部分涉及学习数据结构的基本概念、C语言复习与归纳、递归思想；第二部分涉及线性数据结构、非线性数据结构，包含线性表、栈、队列、字符串、二维数组、树和森林、二叉树、图；第三部分涉及查找、排序等基础应用。为了拓展数据结构的知识，介绍了广义表和文件的基础内容。

相对于C++版的《数据结构与程序构建》，本书的源码进行了重新的设计，相比过去的内容，增加了集合运算、括号匹配判断、文本索引结构、最优二叉树用于哈夫曼编码、查找技术中的平衡二叉树的程序设计源码和歌曲播放等有实用价值的源码，大大提高了本书的源码数量和质量。

本书配有电子教案和程序源代码，便于教师教学和学生使用。本书采用C语言编程实现，程序均在VC++ 6.0下调试运行通过。

数据结构是程序开发的基础，是进入程序设计殿堂的敲门砖。本书是我多年来勤恳敬业、认真教学和仔细思考的结晶，在付出了很多艰辛之后终于有了今天的收获，希望读者能分享和品味学习的乐趣。

我要感谢我的父母，他们给予了我生命和不断进步的力量，也希望在此专门表达对我爱妻的感激，能写出这本教材，也是对她几十年给我的无微不至关怀和帮助的最好回报，在我的心中她是一个完美的女性，还有我诸多朋友的鼓励和支持让我不能忘怀。

我要感谢我多年前在清华大学研习人工智能研究生课程时的指导老师石纯一教授，他对专业的精通、做事的认真以及平易近人的态度让我的一生都受益匪浅，那段时光是我一生的回忆和骄傲。

我非常感谢我的领导和同事，在我多年讲授数据结构过程中给予我很多的支持和鼓励，感谢周海鹰教授、陈宇峰教授、史旅华教授。

本书能够顺利的出版，得益于清华大学出版社广大员工的支持和鼓励，以及细致的审稿和深入全面的建议，深深地感谢他们。

我要感谢多年来参与程序编写测试、文字校对的学生团队，他们当中有许多我的得意弟子，如何清、陈帅、赵伟、熊锐等。

本书第2章由付勇智老师编写，其他章节由马春江老师编写。

本人的电子邮件地址为1932196@qq.com，各位老师可以来信联系程序源码和其他教学资料，希望读者提出宝贵的意见和建议，以便再版时改进和提高。

马春江　于湖北汽车工业学院

2014年12月31日

目　录

第1章 数据结构基础

本章介绍数据结构的基本背景和基本概念，并复习高级语言中诸多重要的基础知识，为数据结构相关的程序设计奠定基础；讲解逻辑结构分类、存储结构分类与基本操作的名称和特点，讨论算法和算法效率分析，最后给出一个高级语言综合复习的程序源码，供读者熟悉相关的程序设计技巧和本书的程序源码设计风格。

1.1 面式思维和点式思维

读者在学完高级语言(如C或C++)程序设计之后通常希望能尽快展示出自己的程序设计能力，开发出一些原创的软件作品，但却发现很多程序设计很难动手，一开始就陷入了不知所措的情形，发现自己好像根本不会编程？这是一个令人困惑的问题。

下面通过一个问题的讨论来看看这究竟是怎么回事。

在图1-1所示的模拟白板上有一批数据，请问哪一个是最大值？

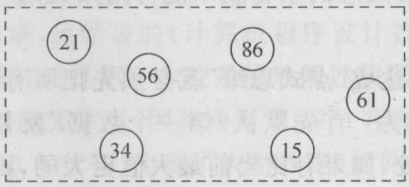

图1-1 面式思维的原理示意图

很显然，读者一眼就可以说出最大值是86，那么深入追究一下，读者根据什么说最大值是86？整个观察、判断、思考的过程是什么？能否书写出完整的思考过程呢？

读者可能会直接回答说“比较”呗，那么最先比较的是哪两个数据？为什么要先比较那两个数据？如何确保其他数据都能比较到？如何避免重复比较？比较完第一轮后，根据什么选择哪个数据进入下一轮比较？仔细想一想就会发现这些问题是没有固定答案的，很多人甚至说不出来这些思考的过程。

为什么会出现这种情况呢？

这是因为人类有了思维能力之后，经过训练可以达成一种大小比较和选择能力。人的双眼观察世界是立体的，把观察到的信息以图像的方式同时进入大脑，简化这种模型后可以认为已经是平面关系。进一步，人类的大脑开始工作，此时相当于有很多部高速计算机在同时工作，一瞬间结果已经出现了。但是实际上，从开始思考到说出结果这一段的工作过程是无法精确描述的。

把计算机能够做的所有工作称为“计算”，就是一种广义的计算。计算机在进行各种计算时是否和人的思维方式一样呢？

遗憾的是至今为止，人类还没有真正了解人的大脑思维模式，无论是医学家、动物学家，还是心理学家、哲学家，都无法说明人是如何记忆、如何思考的。但是人却提出了让机器来帮助进行计算这样的设想。那么怎样才能让机器做这些事呢？科学家给出了一种思路，那就是“模拟”机制，也就是不管人类是如何完成这些工作的，只要机器最后做出来的结果和人类做的结果一样就可以了。

人类的记忆机制和思考机制虽然不能完全搞清楚，但是可以想象它是一个很复杂的网络结构，复杂到任何一个信息点都可以直接激活另外一个信息点，这就是“联想”，也可以失去任何联系，这就是“忘记”。在这个复杂的网络中，所有信息都是立体的，完成思考和记忆的过程也是立体的，同时人类在观察现实生活中的事物时也是立体的，这些立体信息一瞬间同时进入大脑，之后人类开始立体式思维过程和记忆过程。从上面求最大值来看，眼睛把所有数据一次看完，之后同期进入大脑，开始立体式网状思维。本书把这种思维方式称为“面式思维方式”（指的是多面组成的网状结构），简称“面式思维”。

对应记忆方面，计算机中使用的是“存储器”，而对应思考方面，计算机中则使用的是“中央处理器”（CPU），它的特点是在某个时刻永远只能处理当前特定“一个”存储单元的信息。

虽然内存中可以同时存放很多数据，但是中央处理器实际上工作在具体的“点”上，而不是线条、平面图形、立体图形等。基于此点把计算机的思维方式称为“点式思维”，那么编程实际上就是把“面式思维”转化为“点式思维”的过程，或者说如何用“点式思维”来模拟“面式思维”，所以在编程中对任何要处理的事物就不能再用人类的一般思维方式而应该习惯去用计算机的思维方式。

针对求最大值问题，为了达成“点式思维”就必须先把所有数据排成一行（或一列），然后确定求最大值的规则，其思路为：首先默认“第一个数据”就是最大值，然后把后面的数据“逐一”和当前最大值进行比较，如果有比当前最大值更大的，则记录该位置，设其为最大值，继续比较过程，直到比完“最后一个数据”，最后就求出了最大值的位置。图 1-2 为重新排列后的图 1-1 中的那批数据，在上述讨论中，“第一个”，“逐一”，“最后一个”等字眼就是点式思维最明确的象征，而在图 1-1 当中，这些概念都是无法体现的。通过这样的讨论大家就会有一个统一的结论。

图 1-2　点式思维的原理示意图

这里的“一行”实际上就是第 3 章将要介绍的数据结构“线性表”，而算法的名称可以说是“顺序比较与刷新法求最大值”。

明白了“点式思维”的基本特点后，程序设计的很多思路都将变得相对简单，而在数据结构的程序设计中更是充满了“点式思维”的影响。

1.2 数据结构背景

计算机发展初期，人们使用计算机主要是处理数值计算问题，如天气预报、军事领域、大型工程等计算量大的工作，通常解决这样的一个具体问题需要经过几个步骤：首先从具体问题抽象出一个数学模型，然后设计或选择一个求解此数学模型的算法，最后编写程序进行调试，直至得到最终的结果。

数值计算问题的特点是数据之间的关系不太复杂，容易处理。数据量并不大，但是计算量巨大。其主要工作是努力提高运算速度和精度。早期所涉及的运算对象是简单的整型、实型或布尔类型数据，编程者的精力主要集中在程序设计技巧上，无须重视数据结构。

随着计算机应用领域的扩大和软、硬件的发展，非数值计算问题变得越来越普遍和重要。据统计，当今处理非数值计算性问题（如数据处理、图形处理等）占用了计算机90%以上的运行时间。这类问题涉及的数据结构相当复杂，数据元素之间的关系已经无法用数学方程式加以描述，故非数值计算问题的特点是数据量巨大且关系复杂，但是计算量却不大。

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。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。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。

数据结构作为一门独立的课程在国外是从1968年开始的，在此之前其有关内容已散见于编译原理及操作系统等课程之中。20世纪60年代中期，美国的一些大学开始设立该课程，但当时的课程名称并不是数据结构。1968年美国的唐·欧·克努特教授（Knuth D. E.）开创了数据结构的最初体系，他编著的《计算机程序设计技巧》（《The Art of Computer Programming》）第一卷《基本算法》是第一本较系统地阐述数据逻辑结构和存储结构及其操作的著作，可以说是前无古人，后无来者，因此获得了1974年图灵奖。

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发展，如多维图形数据结构、并行计算数据结构等；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，并越来越被人们重视。

数据结构发展史上还有一位不能不提的巨奖，就是许多年前风光无限的PASCAL语言的开发者尼克劳斯·沃思（Wirth N），他撰写的《算法＋数据结构＝程序》（《Algorithms＋Data Structures＝Programs》）依然是具有里程碑意义的优秀书籍之一。他是结构化程序设计的首创者，是1984年图灵奖的获得者。即使现在面向对象已大行其道，在对象中的函数实现时，依然可以体会到这句名言的价值。

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。在进行科学计算、数据处理、过程控制以及对文件的存储、检索及数据库技术等计算机应用领域中，都有对数据进行处理的过程。因此，要设计出一个结构良好、效率很高的程序，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，进一步编出程序。

数据结构技术还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。数据结构是计算机相关专业的专业基础课，是十分重要的核心课程。所有的计算机系统软件和应用软件都要用到各种类型的数据结构。因此，要想更好地运用计算机来解决实际问题，仅掌握高级语言是不够的。要想有效地使用计算机，充分发挥计算机的性能，开发出规范、高效、实用的软件，则必须掌握数据结构的知识。

数据结构课程内容非常丰富，而且与数学、计算机硬件和其他软件都有十分密切的关系。可以认为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，是编译原理、操作系统、数据库原理、软件工程、人工智能等后续课程不可或缺的基础。

学习数据结构的目的是为了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逻辑思维能力，通过程序设计技能训练来促进综合应用能力和专业素质的提高。

程序设计的基本理论涉及三门重要的基础课：高级语言程序设计、数据结构、算法设计与分析。高级语言程序设计主要是解决上机编程的环境、语法要求和一些基本的程序设计技巧，数据结构主要是把数据的关系研究清楚并且确定其存储方案，以及基本操作的编程实现。算法设计与分析主要是深入研究算法的设计技术和时间空间效率分析，这三者可谓缺一不可。在算法设计与分析中会把时间效率分析重点讨论，故本书中没有将时间复杂度的计算过程与深入分析作为重点，仅仅通过一些简洁分析确定其结论。

数据结构课程主要讨论软件开发过程中的设计阶段，同时设计编码和分析阶段的若干基本问题也会被综合讨论。此外，为了构造出好的数据结构并实现，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3个层次5个“要素”，如表1-1所示。

表1-1　数据结构研究的主要内容

方面 / 层次	数据表示	数据处理
抽象层	逻辑结构	基本操作
实现层	存储结构	算法
评价层	各种存储结构的比较及算法分析	

1.3　数据结构的应用案例

在学习程序设计的过程中，必须体会到“量变引起质变”的重要性。当面临的任务或软件系统足够复杂时，很多无足轻重的“小事”都会演变成“大事”，那么整体设计就成为首要的工作。在使用高级语言编程时如果不进行整体设计，边想边做，也不按照开发规范书写，一旦该开发任务涉及的数据之间的关系比较复杂，就可能出现多次返工或彻底开发失败的结局，所以在软件开发前必须首先考虑好各种情况，尤其是数据的关系（即数据结构）的设计。

【应用案例1-1】 图书馆管理计算机化。没有计算机之前，图书馆就已经存在。如果

把成千上万的图书直接放在一个超大的礼堂中，借书时进去一本一本地翻找。即使有耐心，也许花几天时间也很难找到，而通过管理情况就发生了变化。现实中图书馆管理都是把图书进行分类、编号，再按照楼层，房间，书架号等信息进行放置。在读者提出借书要求后，只要有书号，就能在很短时间里找出该书，这就是管理。但是能很快找到该书，却不一定能提供更深入的信息，如该书有几本？被谁借去了？什么时候能还回来？图书馆中某个著者编著的所有书有哪些等。而进一步启用计算机管理后，这些问题都容易解决，但是对于成千上万的数据，如果需要数个小时甚至数天的时间才能得到查询结果，即使这个结果是准确的，也会因为失去了时效性而降低了实用价值。本案例说明数据必须经过组织和管理才能高效地发挥作用，同时即使功能正确但是运行效率太低也没有太大的实用价值，故必须学习利用数据结构的知识提高查找操作的时间效率。

【应用案例 1-2】 人机对弈(如下棋等)程序设计。有些读者喜欢计算机可能是因为利用计算机可以玩一些娱乐性的游戏，而作为专业工作者需要更深层面的思考。例如，各种棋类的对弈是如何编程实现的，首先是界面的实现，如棋盘和棋子如何实现，显然不能是简单的图片，因为还要考虑棋子的移动。另外，要对弈的话，计算机怎么知道哪些位置可以落子，对方的哪些棋子已经被吃掉，而自己的哪些棋子已经受到威胁，进一步又如何决策进攻或防守呢？还有其他更多的设计细节都是一个个难题。通过这个案例可以认识到不论是界面还是功能都会涉及数据结构的大量知识，如果没有数据结构很难编出游戏或更复杂的软件。

【应用案例 1-3】 图形处理软件的开发。如果用图形处理软件如 Photoshop 来处理图片，就会被它神奇的功能所震撼，因为可以轻易地对图片进行放大、缩小、移动、旋转、剪裁、拼接、变色等操作。实现这些功能看上去很复杂，底层实现却较简单，这就是数学的矩阵运算。如果面临巨大的矩阵相乘，如何存储，又如何运算呢？因为希望很快能看到变形后的效果，这时就很敏感程序运行的时间，那么有什么好的加快速度的方法呢？面对巨大的数据量，又有什么好的存储方法能节省存储量呢？这个案例说明了数学原理对于程序设计非常有价值，使用计算机处理大量数据时如何提高存储效率和处理速度是很困难的，因此数据结构的学习也非常重要。

【应用案例 1-4】 计算机计算表达式。计算机能计算表达式，似乎天经地义，否则为什么叫计算机呢？实际上使用很简单，但当初科学家实现这个功能时却很困难。试想遇到 2+1，计算机能开始计算吗？不行，因为可能是 2+11，所以必须往后读到回车或者另外一个运算符，但是遇到运算符还是不能做，因为可能是 2+11×3。根据运算规则，必须先计算 11×3，可是表达式又可能是 2+11×3^2，那么能不能先算平方呢？也不能，因为原来的表达式可能是 2+11×3^(2+1)，既然有括号，而括号中又出现了加法，那么最初的问题就会重新出现，如 2+11×3^(2+1×2)。即使中间的部分计算结果能先求出来，还是有次序问题，如原式为 2+11×3^(2+1×2)−6，那么现在该退回去做加法，还是往前走去做减法呢？已经读过去的数据和运算符如何存储？后面暂时没有读到的数据又如何存储？如果这些数值全部用一些变量存储，那么启用多少个变量？变量之间的关系如何管理？这些问题如果不能解决，想让计算机进行“计算”完全不可能。幸运的是科学家们已经解决了这些问题，使用计算机能够做到数值计算的精确化和高速化，火箭上天、宇宙飞船等大量计算也就有了保证。本范例说明计算机要实现的很多功能仅仅靠编程技巧很难解决，必须依赖数据结构的支撑。

【应用案例1-5】 痕迹检测、DNA检测、文字识别、人像识别等技术的实现前提。现代科技带来了很多前人闻所未闻、想也想不到的科技成果，比如公安系统能够使用指纹或血液等信息追查罪犯、医院能够用DNA做亲子鉴定、计算机还可以进行文字或语音识别等。这些技术面临着一个共同的难题，那就是必须存储超大量的数据，还要进一步进行快速和有效地查找。有时数据量的总量远远超过想象，被称为海量数据。那么面对海量数据，如何存储，数据之间的关系又如何管理，进一步又如何实现快速查找等功能呢？这个范例说明数据结构对数据的组织和查找技术将起到关键的作用。

1.4 数据结构基本概念

(1) 数据(Data)。一般人们谈及数据就会想到0～9组成的数字，实际上这些被称为数值。在现实生活中的“信息”这个概念一旦被计算机存储，就成为所谓的“数据”。要注意的是“数据”并不能完全覆盖“信息”，如日常生活中的“眉目传情”，情感是一种信息但是并不能被计算机识别、存储从而处理，所以数据是信息的一种载体，它能够被计算机识别、存储和加工处理。

计算机科学中，数据可以是数值数据，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；而非数值数据包括字符、文字、图形、图像、语音等，其中的文字又包括英文、中文、中国少数民族的文字、其他各国文字等类型。

(2) 数据元素(Data Element)。数据元素是数据的基本单位。在不同的场合下，可把数据元素称为元素、结点、顶点、记录等。

(3) 数据项(Data Item)。有时一个数据元素还可以由若干个数据项组成。例如，人员信息的每一个数据元素就是一个人的记录，可能包括编号、姓名、性别、籍贯、出生日期、电话、手机、通信地址、电子邮件地址等数据项。这些数据项再次被分为两种：一种叫作基本项，如性别、籍贯等，这些数据项在数据处理时不再分割；另一种叫作组合项，如出生日期可以分为年、月、日等更小的项。

(4) 数据对象(Data Object)。数据对象也可以称为数据元素类(Data Element Class)，是具有相同性质的数据元素的集合。在某些具体问题中，数据元素具有相同的性质，属于同一数据对象，即数据元素是数据对象的一个实例。

(5) 数据结构(Data Structure)。数据结构是指互相之间存在一种或多种关系的数据元素的集合。在使用计算机处理任何问题时，我们不是仅仅关注数据本身，而一定会关注数据元素之间的关系，因为这些“关系”就是我们需要的“信息”，这种数据元素之间的关系就是所谓的“结构”。

(6) 数据结构的形式。数据结构是一个二元组

$$Data\_Structure = (D, R)$$

其中，D(Data的头字母)是数据元素的有限集，R(Relation的头字母)是D上关系的有限集。

(7) 数据的逻辑结构。数据的“逻辑结构”可以看作是从具体问题抽象出来的数学模型，是数据本身的关系而已，它与数据的存储并无关系，有点像纸上谈兵。但是研究数据结