HZ BOOKS
华章教育

高等院校计算机专业人才能力培养规划教材

数据结构
C语言描述

殷人昆 编著

KNOWLEDGE · INNOVATION · CAPABILITY

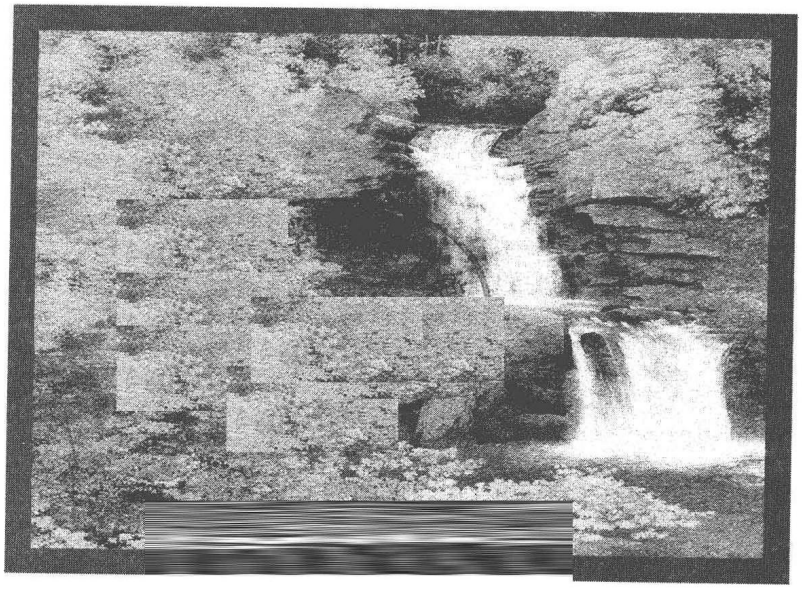
Data Structures in C

机械工业出版社
China Machine Press

高等院校计算机专业人才能力培养规划教材

数据结构
C语言描述

殷人昆 编著



Data Structures in C

机械工业出版社
China Machine Press

本书是根据2007年教育部颁发的《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》规范和2010年修订的《全国硕士研究生入学统一考试计算机专业基础综合考试大纲》编写的数据结构课程教材。全书共分7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，算法分析的简单方法，以及用C语言编程的要点；第2～7章对应考试大纲的6个知识单元，包括线性表，栈、队列和数组，树与二叉树，图，查找，排序。本书融入了作者三十多年的教学经验和考试辅导体会，合理安排相关知识点，对学生容易忽略的地方和隐含在所讨论问题之后的内容给出适当的提示。

本书既可作为普通高校计算机科学与技术及相关专业本科生学习数据结构课程的教材，也可作为计算机专业考研的辅导教材或其他专业计算机考试的复习教材，还可作为计算机系统开发人员的学习资料。

图书在版编目（CIP）数据

数据结构：C语言描述/殷人昆编著. —北京：机械工业出版社，2011.6
（高等院校计算机专业人才能力培养规划教材）

ISBN 978-7-111-34971-6

Ⅰ. 数… Ⅱ. 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

中国版本图书馆CIP数据核字（2011）第103018号

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：刘立卿
北京市荣盛彩色印刷有限公司印刷
2011年6月第1版第1次印刷
185mm×260mm · 20印张
标准书号：ISBN 978-7-111-34971-6
定　　价：35.00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
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

出版者的话

机械工业出版社华章公司多年来以“全球采集内容，服务中国教育”为己任，致力于引进国际知名大学广泛采用的计算机、电子工程和数学方面的经典教材，出版了一大批在计算机科学界享誉盛名的专家名著与名校教材，其中包括 Donald E. Knuth、Alfred V. Aho、Jim Gray、Jeffery D. Ullman 等名家的一批经典作品。这些作品为我国计算机教育及科研事业的发展起到了积极的推动作用。

近年来，我们一直关注国内计算机专业教育的发展和改革并大力支持、参与相关的教学研究活动。2006 年，教育部高等学校计算机科学与技术专业教学指导分委员会在对我国计算机专业教育现状和社会对人才的需求进行研究的基础上，发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》）。为配合《规范》的实施和推广，我们出版了“面向计算机科学与技术专业规范系列教材”。这套教材的推出，对宣传《规范》提出的“按培养规格分类”的理念、推进高校学科建设起到了一定的促进作用。

2007 年，教育部下发了《关于进一步深化本科教学改革全面提高教学质量的若干意见》，强调高等教育以育人为本，以学生为主体，坚持以培养创新人才为重点，下大力气深化教育教学改革。在“质量工程”的思想指导下，各高校纷纷开展了相关的学科改革和教学研究活动。高等学校计算机科学与技术专业的教育开始从过去单纯注重知识的传授向注重学科能力的培养转型。2008 年年底，教育部高等学校计算机科学与技术专业教学指导分委员会成立了“高等学校计算机科学与技术专业人才专业能力构成与培养”项目研究小组，研究小组由蒋宗礼教授（组长）、王志英教授、岳丽华教授、陈明教授和张钢教授组成，研究计算机专业人才基本能力的构成和在计算机专业的主干课程中如何培养这些专业能力。

为配合“高等学校计算机科学与技术专业人才专业能力构成与培养”专项研究成果的推广，满足高校从知识传授向能力培养转型的需求，在教育部高等学校计算机科学与技术专业教学指导分委员会专家及国内众多知名高校专家的指导下，我们策划了这套“高等院校计算机专业人才能力培养规划教材”。这套教材以专项研究的成果为核心，围绕计算机专业本科生应具有的能力组织教材体系。本套教材的作者长期从事教学和科研工作，他们将自己在本科生能力培养方面的经验和心得融入教材的编写中，力图通过理论教学及实践训练，达到提升本科生专业能力的目标。希望这些有益的尝试能对推动国内计算机专业学生的能力培

养起到积极的促进作用。

华章作为专业的出版团队，长久以来遵循着“分享、专业、创新”的价值观，实践着“国际视野、专业出版、教育为本、科学管理”的出版方针。这套教材的出版，是我们以教学研究指导出版的成功范例，我们将以严谨的治学态度以及全面服务的专业出版精神，与高等院校的老师们携手，为中国的高等教育事业走向国际化而努力。

华章教育

高等院校计算机专业人才能力培养规划教材

编 委 会

丛书序言

作为我国规模最大的理工科专业，计算机本科专业为国家的建设培养了大批人才。2006年，教育部计算机科学与技术专业教学指导委员会发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》），提出了以“按培养规格分类”为核心思想的专业发展建议，把计算机专业人才划分为研究型、工程型、应用型3个类型。在《规范》的方针指导下，培养合格的计算机本科人才。

教育包括知识、能力、素质三个方面，专业教育不仅要重视知识的传授，更应突出专业能力的培养，实施能力导向的教育。如何以知识为载体实现能力的培养和素质的提高，特别是实现专业能力和素质的提高是非常重要的。对计算机专业本科教育而言，要想实现能力导向的教育，首先要分析专业能力的构成并考虑如何将其培养落实到教学实践中。为此，教育部高等学校计算机科学与技术专业教学指导分委员会开展了计算机科学与技术专业人才专业能力（简称计算机专业能力）的培养研究。该项研究明确了计算机专业本科人才应具有的4大基本能力—计算思维能力、算法设计与分析能力、程序设计与实现能力、系统能力，并将这四大基本能力分解为82个能力点，探讨如何面对不同类型学生的教育需求，在教学活动中进行落实。

为体现研究成果在教学活动中的实现，我们根据《高等学校计算机科学与技术专业人才专业能力构成与培养》，出版了这套教材。本套教材面向高等院校从知识传授向能力培养转型的需求，在内容的选择、体系安排和教学方法上按照专业能力培养的需要进行了探索。其主要特点有：

（1）以教学研究为先导。本套教材以计算机专业能力专项研究成果为基础，体现了先进的教育理念和教学方法，内容选择、知识深度、结构安排更加符合计算机专业教育的需求。

（2）落实能力培养的思想，同时满足课程的要求。本套教材不仅关注知识点的讲授，还凸显能力培养的要求，将能力的培养分解到各门课程的各个知识点的讲授中。

（3）力求贴近教学实际。作者均长期从事实际教学工作且对专业能力培养具有一定研究，教材编写注重科学组织内容、合理安排体系、便于教学实施，更具操作性。

（4）构建立体化教材。为了方便教师的教学活动，配合主教材开发配套的实验教材、教师参考书、学生辅导书、电子课件等教辅资源。

本套丛书的出版是在配合计算机专业能力的培养和落实方面的初步尝试，我们衷心希望本套教材的出版能起到抛砖引玉的作用，也希望广大教育工作者加入到能力培养的研究和实践中来，并对相关的教材建设提出自己的宝贵意见。

丛书编委会

前　言

从2006年到2007年，为适应计算机科学与技术专业人才培养的需求，教育部高等学校计算机科学与技术教学指导委员会颁发了《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》规范，该规范参照ACM、AIS和IEEE-CS发布的Computing Curricula 2005，从问题空间、知识取向、能力要求等方面，给出了教育改革和专业能力培养的新要求。按照知识取向的不同，把计算机科学与技术学科划分为计算机科学（CS）、计算机工程（CE）、软件工程（SE）和信息技术（IT）四个方向。从问题空间划分，计算机科学属于科学型，计算机工程和软件工程属于工程型，信息技术属于应用型。

规范中规定，数据结构课程涵盖的知识单元有5个：基本算法（AL3）、算法与问题求解（PF2）、基本数据结构（PF3）、递归（PF4）和分布式算法（AL4）。教学大纲包括：（1）算法、算法的时间复杂度和空间复杂度，最坏和平均时间复杂度等概念；（2）算法描述；（3）常用算法设计方法：迭代法、穷举搜索法、递推法、算法的递归描述技术；（4）数据结构的基本概念和术语：数据结构、数据类型、抽象数据类型、信息隐藏；（5）线性表、线性表的存储结构：顺序、链接；（6）栈、队列；（7）串；（8）多维数组和广义表；（9）树形结构及其应用：树、森林、二叉树、线索二叉树、Huffman树；（10）图及其应用：图的基本概念、图的存储结构、图的遍历、生成树和最小生成树、最短路径、拓扑排序；（11）常用排序算法：插入排序、交换排序、选择排序；（12）常用查找技术：线性表上的查找、树的查找、散列技术；（13）文件：顺序文件、索引文件、索引顺序文件、散列文件。

不同的学科方向对以上知识单元的侧重点有所不同，见下表。

学科方向	涵盖知识点	学时
计算机科学	算法分析基础、算法策略、基本算法、分布式算法、可计算性理论基础、算法与问题求解、基本数据结构、递归	48+16
计算机工程	基本算法分析、算法策略、计算算法、分布式算法、算法复杂性、算法与问题求解、基本数据结构、递归	48+16
软件工程	计算机科学基础（程序设计基础、算法及数据结构/表示、抽象（使用和支持）程序设计语言基础）、测试（例外处理）	48+16
信息技术	基本数据结构、算法和问题解决、递归	48+16

在2008年，教育部决定从2009年起，对全国硕士研究生入学统一考试计算机科学与技术学科的初试科目进行调整，其中对（工科）专业课采取了全国统一命题、统一考试。数据结构课程被列为考试科目之一。从3年批改考卷的经验来看，考试范围基本没有脱离考试大纲，也没有超出规范的要求。

然而，从考生的表现来看，答题的水平相差太多。2011年北京8000多名考生参加考试，选择题80分，平均得分48分；综合应用题70分，平均得分30分。这说明很多考生学得不够深入、扎实。为此，作者决心基于多年的教学经验和知识积累，编写一本适合广大有志于学好数据结构课程的学生学习的教材。

本书以专业基础能力培养为目标，承续计算机程序设计基础课程，完全遵照《全国硕士研究生入学统一考试计算机科学与技术学科联考考试大纲》，并参考教育部计算机科学与技术教学指导委员会《高等学校计算机专业人才专业能力构成与培养》的要求编写而成，旨在培养学生的基本计算思维能力，提高学生的算法设计和程序实现能力，并为学生提高系统开发能力打下良好的基础。为此，在编写本书的过程中做了如下工作：

（1）采用“工科”思维，启发学生掌握“化复杂为简单”的方式，从问题入手，通过“问题/子

问题”分解，寻找解决方案。

（2）对基本知识点讲深讲透，通过多种应用举例，让学生了解不同问题需要采取什么方法来应对。

（3）通过大量习题，从不同视点、不同层面训练学生，培养其联想能力，提高其分析问题、解决问题的能力。

（4）配合辅助教材《数据结构习题精析与考研辅导》[⊖]。辅助教材提供了600多题的参考答案和解析，并就关键点进行点拨，另外，提供了多套模拟题。

也就是说，本书的宗旨是培养学生从对问题的理解入手，运用已掌握的算法和数据结构知识寻找解决途径，构建适当的算法和基本程序，以求得问题的解决。在从简单到复杂讨论数据结构的过程中，逐步引入常用的算法设计策略，通过编程上机实验到工程实践，使学生初步掌握分析问题、解决问题的方法。

全书共分7章，各章内容如下：

第1章介绍了数据、数据结构及其算法等基本概念与基本知识，以及学习数据结构所需要掌握的C语言程序设计预备知识（这部分内容考试大纲未列出，但它为全书打下了基础，其中有一些概念在2011年的试卷中出现过）。

第2章介绍了线性结构及其基本操作内容，并给出基于数组与链表表示的情况下，其基本操作函数的实现方法。在介绍线性表的应用时，讨论了集合、字符串和多项式的实现。

第3章介绍了栈、队列、数组和双端队列的常用表示方法，以及栈、队列和数组的应用举例，还讨论了递归、分治、回溯、动态规划等算法设计的策略。

第4章介绍了树与二叉树的概念、表示方法及其常用操作的实现，还介绍了线索二叉树等特殊的二叉树结构，树与森林的常用表示方法及其基本操作的实现方法，二叉树与树的应用，包括二叉排序树、平衡二叉树、Huffman树、堆等。特别地，本章借讨论高斯八皇后问题，进一步讨论了回溯法，并引入了状态树和剪枝方法。

第5章介绍了图的概念、图的基本表示方法及其常用操作的实现，还介绍了图的遍历以及连通图、重连通图，此外还介绍了图的应用，包括生成树和最小生成树、最短路径、拓扑排序、关键路径等，多次涉及贪婪法等算法设计策略。

第6章介绍了多种常用的查找算法，包括顺序查找、折半查找算法及其扩展、索引顺序查找（分块查找）、多路查找树、B树、B^+树、散列表等。

第7章介绍了多种常用的排序算法。

全书采用C语言作为数据结构及算法的描述工具，适当采用C++的少量语句以简化程序。算法描述力求结构化，注重编程风格，每个算法基本保持在100行之内，可读性强。

本书可以作为计算机科学与技术及相关专业本科生的教材，也可以作为计算机专业考研（硕士、工程硕士、博士）的复习教材，还可以供使用计算机做系统开发的人员学习使用。

本书是从作者多年在清华大学讲授“数据结构”课程的课件取材，并经过一定改编而成，因此在叙述上适合学生理解，在内容上浅显易懂，在选材上舍弃了很多超出规范和大纲的内容，增加了许多在一些老教材中忽略了的内容，希望不会辜负读者的期望。

各章所附习题不包括选择题，但精选了大量综合应用题，这些习题的参考解答请参看配套教材《数据结构习题精析与考研辅导》。

由于作者的水平有限，可能在某些方面考虑不周，书中难免存在疏漏或错误，恳请读者提出宝贵意见。

作者邮箱：yinrk@tsinghua.edu.cn或yinrk@sohu.com。

编　者

2011年4月于清华园荷清苑

⊖ 本书已由机械工业出版社出版，ISBN 978-7-111-32283-2，定价：45.00元。——编辑注

教学安排建议

周次	教 学 内 容	学时	程序上机实习/大作业
1	数据结构的基本概念，算法及其特性，算法设计过程和设计策略，算法的简单分析与时间/空间复杂度	3	①统计执行频度 ②出错处理：报错方式
2	线性表，顺序表及其操作的实现，顺序表应用，单链表及其操作的实现	3	用数组实现集合及其常用操作
3	单链表的应用，循环链表，双向链表，字符串及其实现，多项式及其运算的实现	3	用有序链表实现集合及其常用操作
4	栈的概念与实现，队列的概念与实现，双端队列的概念，递归与递归栈	3	火车调度车厢排列 大作业：离散事件模拟——银行队列
5	汉诺塔与分治，迷宫与回溯，组合与动态规划，多维数组与存储，特殊矩阵压缩存储	3	实现迷宫算法，并对递归与非递归算法进行比较
6	稀疏矩阵的表示与相加，树与森林的概念，二叉树性质，二叉树的存储表示	3	魔方问题
7	二叉树遍历，树与森林存储表示，树与森林遍历	3	①递归建立二叉树 ②递归打印二叉树
8	线索二叉树，二叉排序树	3	①建立二叉排序树 ②实现查找/插入/删除算法
9	平衡二叉树，Huffman 树与应用，堆与优先级队列的概念，堆的建立、插入和删除	3	堆的建立、插入与删除 大作业：Huffman 算法——编码/译码
10	图的概念，图的存储表示，图的遍历算法，图的连通性	3	①建立图的邻接表 ②层次序遍历
11	重连通图，最小生成树，拓扑排序	3	①用避圈法构建最小生成树 ②用破圈法构建最小生成树
12	关键路径，最短路径，查找的概念	3	大作业：建立求解欧拉回路/哈密尔顿回路问题的标准函数
13	顺序查找，折半查找，斐波那契查找，插值查找，索引顺序查找，多路平衡查找树	3	①实现斐波那契查找算法 ②实现插值查找算法
14	B 树，B^+ 树，散列表	3	
15	排序概述，直接插入排序，折半插入排序，起泡排序，简单选择排序，希尔排序，快速排序	3	①测定希尔排序性能 ②测定快速排序性能
16	堆排序，归并排序，基数排序，排序方法的下界，排序方法的比较，链表排序	3	①测定归并排序性能 ②测定堆排序性能

说明：

1）建议课堂教学 48 学时，并全部在多媒体教室内进行。

2）建议上机和工程实践（大作业）总学时为 16，如果不够，可在课外追加。学生尽可能使用自备电脑。

3）建议上 3 次习题课，可安排在第 4 周、第 10 周和第 15 周，总结学生在作业和上机过程中遇到的问题，总结解题的类型及解题的方法。

目　录

第1章

Chapter 1

绪　论

【本章学习要点】

- 数据、数据元素、数据结构、数据的逻辑结构和存储结构、数据类型的概念
- 算法的定义及算法的特性
- 算法设计的方法和策略、算法设计的过程、评判算法优劣的标准
- 使用 C/C ++ 语言描述数据结构和算法的方法
- 算法的性能分析：算法的性能标准、算法的后期测试、算法的事前估计，空间复杂度度量、时间复杂度度量、时间复杂度的渐近表示法，渐近的空间复杂度

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

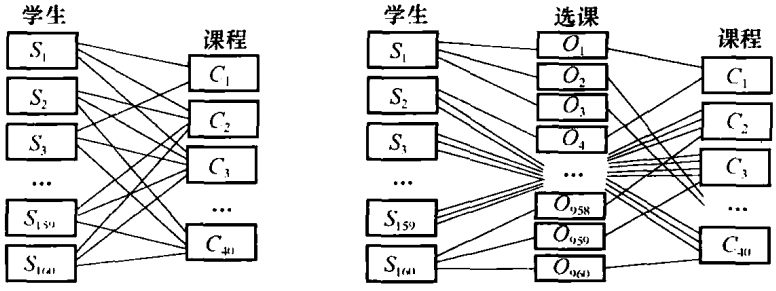
人们在使用计算机解决一个具体问题时，通常遵循的步骤是“理解需求—分析建模—设计解决方案—编程实现”。理解需求好比我们考试，面对一个问题，首先要弄懂这个问题需要你做什么。分析建模就是把你对问题的理解按不同视角进行整理，用模型进行抽象。其中最重要的是两个模型：一个是处理模型，描述问题求解的流程；另一个是数据模型，描述问题所涉及的数据实体和数据实体的组成。设计解决方案将根据已建立的模型，提出求解的算法，并基于算法设计适当的数据结构和程序框架。编程实现将把程序框架用某种程序设计语言翻译成计算机可执行的程序，经过测试和修改，最终得到适用的程序。

例如，对高层建筑进行结构分析，目的是了解高层建筑在外来载荷下的应力应变情况，考查建筑设计方案的可行性，这就是需求（第一步，理解需求）。第二步，根据地质、水文、气流以及建筑材料的特性，建立结构的数学模型。为此需借助有限元法，把整个建筑划分成网格，在每个网格结点上列出变分方程，将变分方程进行一系列变换，最后化为包括成百上千个方程的大型线性方程组。第三步，这个方程组是一个大型稀疏系数线性方程组，其系数矩阵是一个对角块矩阵，可考虑使用分块处理的高斯消去法求解，避免存储大量的零元素。第四步，用 FORTRAN 语言或 C 语言定义相应算法和数据结构，并在选定的计算机上实现。

这属于数值计算的问题。类似地还有利用差分格式分析核爆冲击波的衰减情况、利用基尔霍夫定律进行电路分析等。此外，还有一类属于非数值计算的问题。例如，开发学生选课的系统就是此类非数值计算的问题。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。这样，学生和课程之间出现了多对多的关系，如图 1-1a 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 1-1b 所示。

为解决以上问题，首先理解需求。每学期开学前，为计算机系的学生设置可选课程的列表，学生在开学的前两周可以选课和退选。每位学生限定最多一学期选 6 门课，每门课程限定最多可接受 40 名学生。第二步，分析建模。处理模型包括选课的流程和退选的流程，数据模型包括“学生”、“课程”、“选课”实体的构成和它们之间关系的描述。第三步，设计解决方案，包括设计各个数据实体的数据

结构和存储映像、相应操作的实现逻辑、用户界面的显示和交互的流程。第四步，用C语言或Java语言实现程序代码，用网页制作工具实现菜单树，数据的输入、输出，以及窗口的控制和切换。



a）一个多对多的关系　　b）两个一对多的关系

图1-1　课程与学生之间的两种联系

选课问题是典型的非数值型事务处理问题。类似的问题包括商店销售事务处理问题、电信计费和收费处理问题、电梯控制处理问题、银行排列问题等。

由以上事例可知，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 Structuring》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，我们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（data）

我们在日常生活中会遇到各种信息，如用语言交流的思想、银行与商店的商业交易、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，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（data element）

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（data item）

数据元素可以是单个元素（称为原子），也可以由数据项构成。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成绩、化学成绩等更小的项。

4. 数据结构（data structure）

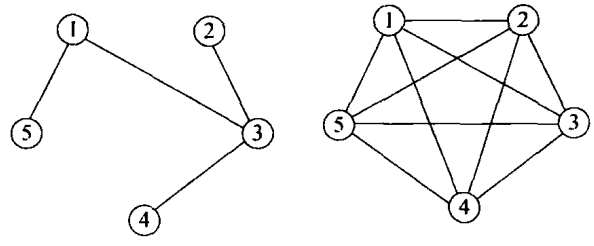
在数据处理中所涉及的数据元素相互之间都不是孤立的，而是存在着某种关系，这种数据元素之间的关系称为结构。例如，招生考试时把所有考生按考试成绩从高到低排队，所有考生记录都将处在一种有序的排列中。又例如，在n个网站之间建立通信网络，要求以最小的代价将n个网站连通，如图1-2a所示，这样，在所有网站之间形成一种树形关系；反之，要求网络中任一网站出现故障，整个

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 1-2b 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。

5. 数据对象（data object）

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组值的集合，且数据对象应是数据抽象和处理抽象的封装体，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。



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

6. 数据类型（data type）

从程序设计角度来看，数据类型与数据对象的概念是相通的，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，整数类型是由范围为 -2^{15} ~ $2^{15}-1$ 的整数构成，并包括对这些整数的加、减、乘、除和取模运算等。

数据类型按照其值的复杂程度不同，可分为基本数据类型和结构数据类型两种。基本数据类型中的每个数据元素都是无法再分割的整体，如一个整数、浮点数、字符、指针、枚举量等，故称为原子类型。结构数据类型由基本数据类型或子结构类型按照一定的规则构造而成。例如，一个学生的学籍卡片是一个结构数据类型，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。

在如 C 或 Pascal 这样的程序设计语言中，不但规定了一些基本的数据类型，还提供了一些构造组合类型（如数组型、构造型、文件型等）的规则。程序员可以利用这些规则，自行定义为解决应用问题所必需的数据类型。因此，数据类型是确切地描述数据对象，正确地进行相关计算的有效工具。例如，在 C 语言中对一个数据表（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 int ElemType;                        //表中元素的数据类型,假定为 int
typedef struct{
    ElemType 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，一些基本数据类型，如 int、float、double、char、string 等，是程序设计语言已经实现了的、程序员可直接使用的数据结构。而结构数据类型是程序员用程序设计语言表示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。

辨析： *有关数据结构的讨论主要涉及数据元素和数据元素之间的关系，不涉及数据元素本身的内容。关于数据元素的内容，在系统开发时考虑。*

7. 抽象数据类型（Abstract Data Type，ADT）

在进行软件设计时，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。在计算机中使用二进制定点数和浮点数实现数据的存储和运算，而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，给出了更高一级的数据抽象，出现了整型、实型、字符型、双精度型等。待到抽象数据类型出现，可以进一步定义出更高级的数据抽象，如各种表、队列、图，甚至窗口、管理器等。这种数据抽象的层次为设计者提供了有力的手段，使得设计者可以

从抽象的概念出发，从整体上进行考虑，然后自顶向下，逐步展开，最后得到所需的结果。

抽象数据类型通常是指由用户定义，用以表示应用问题的数据模型，又称数据抽象。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装在一起，从而实现了更高一级的抽象。

程序1-2给出了自然数（Natural Number）的抽象数据类型定义。

程序1-2 自然数的抽象数据类型

```
ADT NaturalNumber IS
/*objects:自然数是整数的有序子集合,它开始于0,结束于机器能表示的最大整数MAXINT*/
{
    function://对于所有的x,y∈NaturalNumber,+、-、<、==、=等都是可用的服务
    Zero():NaturalNumber                    //返回0
    IsZero(x):Boolean                       //if(x==0)then返回True;else返回False
    Add(x,y):NaturalNumber                  //if(x+y<=MAXINT)then返回x+y
                                            //else返回MAXINT;
    Equal(x,y):Boolean                      //if(x==y)then返回1;else返回0
    Successor(x):NaturalNumber              //if(x==MAXINT)then返回x
                                            //else返回x+1
    Subtract(x,y):NaturalNumber             //if(x<y)then返回0;else返回x-y
}                                           //NaturalNumber
```

程序1-2给出“NaturalNumber”的抽象数据类型描述，objects简要说明数据对象是什么，有什么特点，接下来是一系列可用操作的列表，在操作的说明中要给出前置条件（if）和后置条件（…then…else）。前者给出操作正确执行所需的先决条件，后者表明在前置条件确定后应得到的结果，目的是保证抽象数据类型中每一个操作的正确性。

抽象数据类型最大的特点是把使用与实现分离，实行封装和信息隐蔽。就是说，抽象数据类型有两个视图：外部视图和内部视图。外部视图包括抽象数据类型名称、包含数据对象的简要说明和一组可供使用者使用的操作，这是为了让使用者从外部了解和使用抽象数据类型所必需的。内部视图包括数据对象的存储结构定义和基于这种存储表示的各种操作的实现，包括了一切实现的细节。

现代程序设计方法一个重要的观点是基于对象建立系统，而不是基于功能建立系统。因为系统的功能随着时间的推移必须适应新的情况和要求而不断更新，系统的对象则相对比较稳定。以对象为系统的基本构件搭建的系统比较稳定，每次修改系统功能不会导致全局修改，只需做局部修改即可。在这种情况下，要求构成系统的每一构件一定要满足“封装”和“信息隐蔽”，不允许直接存取构件内部的数据和调用构件内部不对外公开的操作，只能通过构件提供的允许使用者调用的操作来存取构件内保存的数据。这种构件的组织恰恰就是抽象数据类型。

抽象数据类型是属于概念层次的模型，它的实现就是面向对象系统中的“类”，所以，类和对象是抽象数据类型的实现层次的表示。

C++和Java语言都有定义类（class）的机制，可以完全按照抽象数据类型来定义类。所以很多数据结构教材采用C++或Java语言描述。C语言没有可以实现抽象数据类型的相应机制，所以只有先用“typedef struct……”来定义结构类型，再分别定义相关的操作。不过，从教学观点来看，用C语言描述比较简洁，初学者容易入门。

1.1.3 数据结构的分类

1. 分解和抽象

数据结构的核心技术是分解和抽象。首先是数据的分解和抽象，通过分解划分出数据的层次（数据—数据元素—数据项）；再通过抽象，舍弃数据元素的具体内容，得到数据的逻辑结构。其次是处理的分解和抽象，通过分解将处理需求划分成各种功能，再通过抽象舍弃实现细节，得到算法的定义。上述两个方面的结合将问题变换为数据结构和算法。这是一个从具体（具体问题）到抽象（数据结构和算法）的过程。