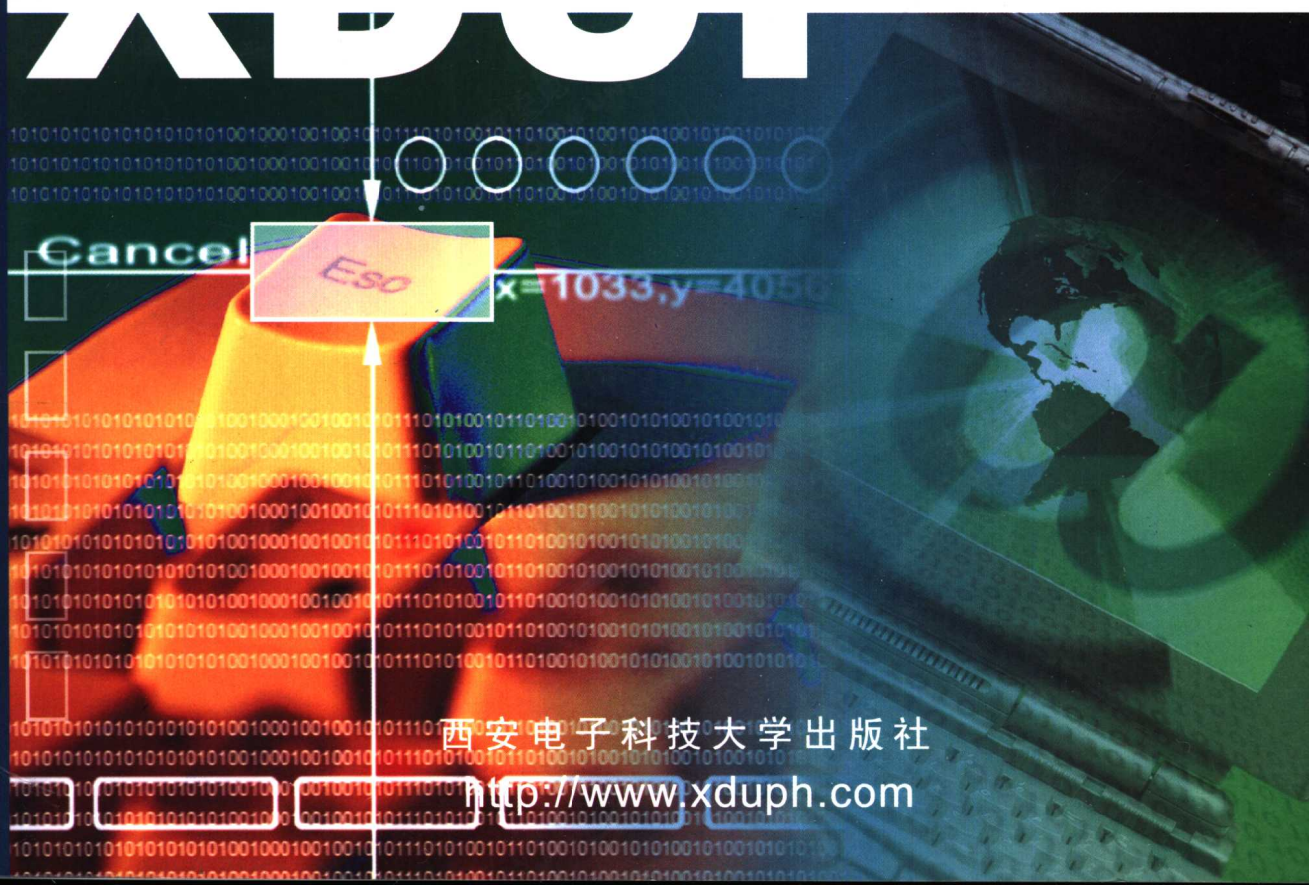
面向21世纪

高等学校计算机类专业系列教材

数据结构—C语言描述

Data Structure in C

陈慧南　编著



西安电子科技大学出版社
http://www.xduph.com

面向21世纪高等学校计算机类专业系列教材

数据结构

——C语言描述

Data Structures in C

陈慧南　编著

西安电子科技大学出版社

2003

内 容 简 介

本书作者基于多年讲授“数据结构”和“算法设计与分析”课程的教学经验，在自己编写并使用多年的用 Pascal 和 C++描述的《数据结构》两书的基础上，参考了近年来国外出版的多种数据结构和算法的优秀教材编写了本书。

本书不仅系统地介绍了各种传统的数据结构和各种搜索及内、外排序方法，还引入了一些比较高级的数据结构，如伸展树和跳表。本书重视算法的时间和空间分析，包括搜索和排序时间的下界分析。书中采用了抽象数据类型的观点讨论数据结构，并使用 C 语言描述。

全书条理清晰，内容详实，既注重数据结构和算法原理，又十分强调程序设计训练。书中算法都配有完整的 C 程序，程序结构清晰，构思精巧。所有程序都已在 TC2.01 下编译通过并能正确运行，它们既是学习数据结构和算法的很好示例，也是很好的程序设计示例。本书内容深入浅出，配有大量的实例和图示，并有丰富的习题，适于自学。

本书可作为高等院校计算机科学与技术专业和其他相关专业的《数据结构》教材，也可供计算机工作者和其他希望学习数据结构和算法知识的人员参考。

★本书配有电子教案，需要者可与出版社发行部联系，免费索取。

图书在版编目(CIP)数据

数据结构：C 语言描述=Data Structures in C / 陈慧南编著.

—西安：西安电子科技大学出版社，2003.8

(面向 21 世纪高等学校计算机类专业系列教材)

ISBN 7-5606-1255-5

Ⅰ. 数… Ⅱ. 陈… Ⅲ. ① 数据结构—高等学校—教材 ② 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(2003)第 045978 号

策　　划　马乐惠

责任编辑　阎　彬

出版发行　西安电子科技大学出版社(西安市太白南路 2 号)

电　　话　(029)8242885　8201467　邮　编　710071

http: //www.xduph.com　　E-mail: xdupfxb@pub.xaonline.com

经　　销　新华书店

印刷单位　陕西光大印务有限责任公司

版　　次　2003 年 8 月第 1 版　2003 年 8 月第 1 次印刷

开　　本　787 毫米×1092 毫米　1/16　印张 19.625

字　　数　462 千字

印　　数　1～4 000 册

定　　价　21.00 元

ISBN 7 – 5606 – 1255 – 5 / TP・0660(课)

XDUP 1526001-1

＊＊＊ 如有印装问题可调换 ＊＊＊

本社图书封面为激光防伪覆膜，谨防盗版

序

第三次全国教育工作会议以来，我国高等教育得到空前规模的发展。经过高校布局和结构的调整，各个学校的新专业均有所增加，招生规模也迅速扩大。为了适应社会对“大专业、宽口径”人才的需求，各学校对专业进行了调整和合并，拓宽专业面，相应的教学计划、大纲也都有了较大的变化。特别是进入21世纪以来，信息产业发展迅速，技术更新加快。面对这样的发展形势，原有的计算机、信息工程两个专业的传统教材已很难适应高等教育的需要，作为教学改革的重要组成部分，教材的更新和建设迫在眉睫。为此，西安电子科技大学出版社聘请南京邮电学院、西安邮电学院、重庆邮电学院、吉林大学、杭州电子工业学院、桂林电子工业学院、北京信息工程学院、深圳大学、解放军电子工程学院等10余所国内电子信息类专业知名院校长期在教学科研第一线工作的专家教授，组成了高等学校计算机、信息工程类专业系列教材编审专家委员会，并且面向全国进行系列教材编写招标。该委员会依据教育部有关文件及规定对这两大类专业的教学计划和课程大纲，对目前本科教育的发展变化和相应系列教材应具有的特色和定位以及如何适应各类院校的教学需求等进行了反复研究、充分讨论，并对投标教材进行了认真评审，筛选并确定了高等学校计算机、信息工程类专业系列教材的作者及审稿人。这套教材预计在2004年春季全部出齐。

审定并组织出版这套教材的基本指导思想是力求精品、力求创新、好中选优、以质取胜。教材内容要反映21世纪信息科学技术的发展，体现专业课内容更新快的要求；编写上要具有一定的弹性和可调性，以适合多数学校使用；体系上要有所创新，突出工程技术型人才培养的特点，面向国民经济对工程技术人才的需求，强调培养学生较系统地掌握本学科专业必需的基础知识和基本理论，有较强的本专业的基本技能、方法和相关知识，培养学生具有从事实际工程的研发能力。在作者的遴选上，强调作者应在教学、科研第一线长期工作，有较高的学术水平和丰富的教材编写经验；教材在体系和篇幅上符合各学校的教学计划要求。

相信这套精心策划、精心编审、精心出版的系列教材会成为精品教材，得到各院校的认可，对于新世纪高等学校教学改革和教材建设起到积极的推动作用。

系列教材编委会

2002年8月

高等学校计算机、信息工程类专业

系列教材编审专家委员会

主　任：杨　震（南京邮电学院副院长、教授）
副主任：张德民（重庆邮电学院通信与信息工程学院院长、教授）
韩俊刚（西安邮电学院计算机系主任、教授）
李荣才（西安电子科技大学出版社总编辑、教授）

计算机组

组　长：韩俊刚（兼）
成　员：（按姓氏笔画排列）
王小民（深圳大学信息工程学院计算机系主任、副教授）
王小华（杭州电子工业学院计算机分院副院长、副教授）
孙力娟（南京邮电学院计算机系副主任、副教授）
李秉智（重庆邮电学院计算机学院院长、教授）
孟庆昌（北京信息工程学院教授）
周　娅（桂林电子工业学院计算机系副主任、副教授）
张长海（吉林大学计算机科学与技术学院副院长、教授）

信息工程组

组　长：张德民（兼）
成　员：（按姓氏笔画排列）
方　强（西安邮电学院电信系主任、教授）
王　晖（深圳大学信息工程学院电子工程系主任、副教授）
胡建萍（杭州电子工业学院电子信息分院副院长、副教授）
徐　祎（解放军电子工程学院电子技术教研室主任、副教授）
唐　宁（桂林电子工业学院通信与信息工程系副主任、副教授）
章坚武（杭州电子工业学院通信工程分院副院长、教授）
康　健（吉林大学通信工程学院副院长、教授）
蒋国平（南京邮电学院电子工程系副主任、副教授）

总 策 划：梁家新
策　　划：马乐惠　云立实　马武装　马晓娟
电子教案：马武装

前　言

有关数据结构与算法的研究是计算机科学与技术的基础性研究之一。美国计算机协会（ACM）和美国电气和电子工程师学会计算机分会（IEEE－CS）在计算学科教学计划CC1991报告中，将算法与数据结构定义为计算学科研究的9个主领域之一。该组织在2001年12月递交的CC2001报告中，调整了计算学科研究领域的划分，最终将计算学科划分为14个主领域。数据结构与算法方面的知识包含在**程序设计基础**（PF）、**算法与复杂性**（AL）和**程序设计语言**（PL）等领域中。

掌握该领域的知识对于我们利用计算机资源，开发高效的计算机程序是非常必要的。因此，CC2001把其中的大多数知识单元规定为计算机及相关学科的本科学生必须掌握的核心知识单元，如**基本数据结构**（PF3），**递归技术**（PF4），**数据类型和数据抽象**（PL4和PL9），**面向对象的程序设计**（PL6），**算法分析的基本方法**（AL1），**基本的计算算法**（AL3）等。其中，基本数据结构包括数组、字符串、堆栈、队列、树、图和散列表等，基本的计算算法包括查找、排序、哈希表算法、搜索树及图算法。目前，国内外所有的计算机专业都开设一到两门相关的课程讲授这方面的知识。

一般来讲，“数据结构”课程涉及的知识单元有PF3、PF4、PL4、PL6、PL9、AL1和AL3。从课程性质来看，“数据结构”是计算机科学与技术专业的一门核心课程，也是计算机软件和应用工作者必备的专业基础。近年来，“数据结构”课程的内容和讲授体系有了很大的变革，在介绍数据结构与算法知识的同时，普遍重视程序设计方面的训练，如C/C++程序设计、Java程序设计等。

本书在内容上注意兼顾广度和深度，不仅系统地介绍了各种传统的数据结构和各种搜索、内外排序方法，还引入了一些比较高级的数据结构，如伸展树和跳表。本书重视算法的时间和空间分析，包括对搜索和排序时间的下界分析。

本书既注重数据结构和算法原理，又十分强调程序设计训练。书中各算法都配有完整的C程序，程序结构清晰，构思精巧。所有程序都已在TC2.01下编译通过并能正确运行，它们对C++环境（如Borland C++、Visual C++）同样适用。这些程序既是很好的学习数据结构和算法的示例，也是很好的程序设计示例。

书中使用C语言描述数据结构。C语言是一种主要的软件开发语言，目前很多高校将其作为第一门程序设计语言讲授。选择C语言描述数据结构，使学习过C语言、C++语言的读者能够很容易地学习本书。对于只学过Pascal语言的读者来说，也可将本书作为学习教材或参考资料。

本书采用抽象数据类型的观点讨论数据结构。面向对象方法是在抽象数据类型的基础上发展起来的软件方法。采用抽象数据类型的观点讨论数据结构与用面向对象的观点讨论数据结构本质上是一致的。学生通过本书的学习，就可以掌握数据抽象的原理。因此，学习本书的读者如果具有C++语言的知识就能方便地阅读用C++语言描述的数据结构书籍。

全书条理清晰，内容详实且深入浅出，书中对算法都做了较详细的解释，尽可能做到可读易懂，并配有大量的实例和图示，有利于读者理解算法的实质和编程思想。各章结尾处的小结可帮助读者了解本章的要点。各章配有丰富的习题，适于自学。

作者多年在南京邮电学院从事“数据结构”和“算法设计与分析”课程教学，曾使用多种教材讲授数据结构和算法，其中包括C语言和C++语言描述的《数据结构》英语原版教材。作者与他人合作编写了《数据结构》(PASCAL 语言描述）教材，1994 年由人民邮电出版社出版。2001 年作者主编了《数据结构——使用 C++语言描述》一书，由东南大学出版社出版。此外，作者还于 1998 年主编了《计算机软件技术基础》一书，由人民邮电出版社出版。本书是在作者上述“数据结构”课程教学和教材的基础上，参考了近年来国外出版的多种数据结构和算法的优秀教材编写而成的。

全书共分 12 章。

第 1 章是预备知识，首先给出传统的数据结构的概念，继而介绍算法规范和数据抽象，最后讨论算法的效率和算法分析的基本方法。

第 2 章介绍两种实现数据的顺序和链接存储的基本数据结构：数组和链表，它们是实现本书中各种抽象数据类型的基础。

第 3、4 章我们定义了几种线性数据结构：堆栈、队列、线性表、数组和矩阵，讨论它们的顺序表示和链接表示，并给出若干应用实例，如表达式计算、多项式的算术运算和稀疏矩阵算法。第 3 章中还讨论了递归和递归过程以及测试数据结构所必需的演示驱动程序的编写方法。

第 5 章简要介绍字符串和广义表的定义、存储表示及典型算法，如字符串匹配等内容。

第 6 章讨论树形数据结构，包括树和森林、二叉树、堆和优先权队列、哈夫曼树和哈夫曼编码、并查集和等价关系等内容。

第 7 章讨论集合和表的搜索，如顺序搜索、二分搜索(包括对半搜索和斐波那契搜索)，搜索算法的二叉判定树以及搜索算法的时间下界等内容。

第 8 章内容包括二叉搜索树、二叉平衡树、伸展树、B 树、键树等。

第 9 章讨论跳表和散列表。

第 10 章讨论图数据结构，着重讨论几种图算法：图的遍历、拓扑排序、关键路径、最小代价生成树和最短路径算法。

第 11 章介绍若干内排序算法。

第 12 章讨论文件和外排序。

附录 A 介绍软件工程的基本概念，包括软件开发方法和系统测试方法。

附录 B 对实习目的、实习要求、实习步骤、实习报告和实习题作了说明和规定。

附录 C 是书中出现的专用名词的中英文对照表。

本书可作为高等院校计算机科学与技术专业和其他相关专业的“数据结构”课程 80 学时的教材。对于学时数少于 80 学时的教学计划，可根据实际学时对本书内容加以剪裁。作者已将难度较大，或非基本的章节标上了*号，供读者参考。

本书的编写得到南京邮电学院和计算机科学与技术系领导的推荐和关心，并得到了西安电子科技大学出版社的支持，在此表示衷心感谢。

书中若有不当之处，敬请读者批评指正。

作　者

2003 年 3 月于南京

目　录

第 5 章 字符串和广义表

第 6 章 树

第 7 章 集合和搜索

第8章 搜索树

第9章 跳表和散列表

第10章 图

第11章 内排序

第 12 章 文件和外排序

第1章

概 论

这一章中，我们首先给出传统的数据结构的概念，继而介绍数据抽象和抽象数据类型的概念，然后给出基于抽象数据类型的数据结构描述方法，最后介绍算法的效率和算法分析的基本方法。

1.1 什么是数据结构

数据结构是计算机科学与技术领域广泛使用的术语，然而，究竟什么是数据结构，在计算机科学界至今没有标准的定义。

随着计算机科学与技术的发展，计算机的应用已远远超出了单纯进行科学计算的范围。今天，信息技术作为现代技术的标志已成为世界各国经济增长的主要动力。计算机技术已渗透到国民经济的各行各业和人们日常生活的方方面面。计算机已经从传统的应用领域，如工业控制、情报检索、企业管理、商务处理、图形图像、人工智能等数据处理领域，发展到了电子政务、电子商务、办公自动化、企业资源管理系统、电子图书馆、远程教育、远程医疗等诸多领域。

现实世界各领域中的大量信息都必须转换成数据才能在计算机中存储、处理。数据是信息的载体。应用程序处理各种各样的数据。笼统地说，所谓**数据**(data)，就是计算机加工处理的对象。数据一般分两类：**数值数据**(numerical data)和**非数值数据**(non-numerical data)。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。非数值数据包括字符、文字、图形、图像、语音、表格等。

数据结构主要是为研究和解决如何使用计算机处理非数值问题而产生的理论、技术和方法。它表达数据的构造形式，即一个数据由哪些成分数据构成，以什么方式构成，具有什么结构。在这里，我们称组成数据的成分数据为**数据元素**(data element)。一般地，数据元素可以是简单类型的，如整数、实数、字符等，也可以是结构类型，如记录。若把每个学生的记录看成一个数据元素，它包括学号、姓名、性别等**数据项**(data item)，一个班的学生记录组成了图 1－1 所示的学生情况表。表是一个数据结构。

从数学概念上讲，一个**数据结构**(data structure)是由数据元素依据某种逻辑联系组织起来的。对数据元素间的逻辑关系的描述被称为数据的**逻辑结构**(logical structure)。

根据数据结构中数据元素之间的结构关系的不同特征，通常将数据结构分为如下四种基本结构：

学　号	姓　名	性 别	其他信息
B02040101	王小红	女	…
B02040102	林　悦	女	…
B02040103	陈菁菁	女	…
B02040104	张可可	男	…
…	…	…	…
⋮	⋮	⋮	⋮

图 1－1　学生情况表

(1) 集合结构(set)：数据元素的有限集合。数据元素之间除了“属于同一个集合”的关系之外没有其他关系。元素顺序是随意的。

(2) 线性结构(linear)或称序列(sequence)结构：数据元素的有序集合。数据元素之间形成一对一的关系。

(3) 树形结构(tree)：树是层次数据结构，树中数据元素之间存在一对多的关系。

(4) 图结构(graph)：图中数据元素之间的关系是多对多的。

图 1－2 给出了上述四种基本结构的示意图。

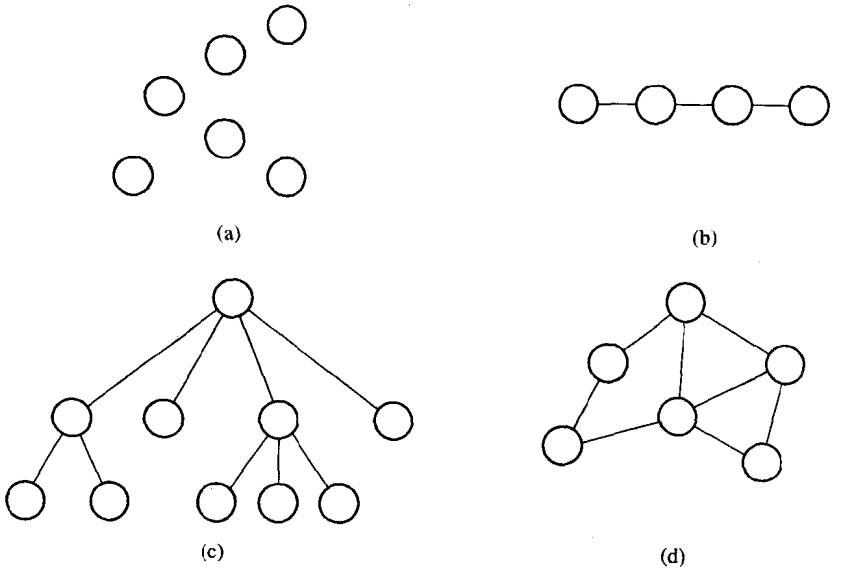


图 1－2　四种基本结构示意图

(a) 集合结构；　(b) 线性结构；　(c) 树形结构；　(d) 图结构

由于集合结构的元素间没有固有的关系，因此往往需要借助其他结构才能在计算机中实际表示该结构。

上述四种基本的结构关系可分为两类：**线性结构**(linear structure)和**非线性结构**(non-linear structure)。我们把除了线性结构以外的几种结构关系——树、图和集合归入非线性结构一类。

数据的逻辑结构是面向应用问题的，是从用户角度看到的数据的结构。数据必须在计

算机内存储，数据的**存储结构**(storage structure)是数据在计算机内的组织方式，是逻辑数据的存储映像，它是面向计算机的。

我们知道，计算机存储器(主存)是由有限个存储单元组成的一个连续的存储空间，这些存储单元或者是字节编址，或者是字编址的。从存储器角度看，存储器中是一堆二进制数据，它们可以被机器指令解释成为指令、整数、字符、布尔数等等，也可以被数据结构的算法解释成为具有某种结构的数据。

顺序(sequential)存储结构和**链接**(linked)存储结构是两种最基本的存储表示方法。

所谓顺序或**连续**(contiguous)的表示方法，也称计算的方法，需要一块连续的存储空间，并把逻辑上相邻的数据元素依次存储在连续的存储单元中。例如有四个数据元素组成的线性数据结构(a_0, a_1, a_2, a_3)，存储在某个连续的存储区内，设存储区的起始地址是100，假定每个元素占2个存储单元，则其顺序存储表示如图1－3(a)所示。

在顺序存储结构表示时，可以容易地用一个数学公式来确定每个元素的位置。对于图1－3(a)的顺序存储结构，一个简单的地址计算公式是：

$$Loc(a_k)=102+2*k \qquad (1-1)$$

公式(1－1)指明了元素 a_k 的存储位置。

这种方法主要用于存储线性的数据结构。对于非线性的数据结构，如树结构，有时也可采用顺序存储的方法表示之。这将在以后讨论。

另一种基本的存储表示方法是链接存储表示。

在链接存储表示下，为了在计算机内存储一个元素，除了需要存放该元素本身的信息外，还需要存放与该元素相关的其他元素的位置信息。这两部分信息组成存放一个数据元素的**结点**(node)。图1－3(b)给出了线性结构(a_0, a_1, a_2, a_3)的链接存储表示。其中每个结点的存储块分成两部分，一部分存放元素自身，另一部分包含该元素逻辑上的**后继元素**(successor)的结点的存储位置。这种关于其他结点的位置信息被称为**链**(link)。

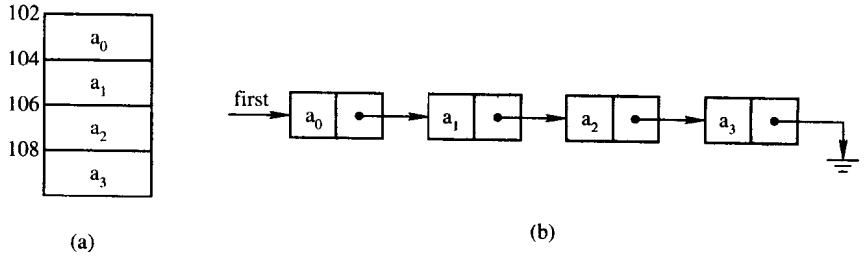


图1－3 两种基本的存储表示方法

(a) 顺序存储结构；(b) 链接存储结构

图(b)中，电接地号表示空链(即不表示任何具体结点的单元地址)。注意：一个结点的存储位置通常是指存放该结点的存储块的起始单元的地址。

在以后的讨论中，在不会引起混淆的场合下，我们可以混合使用结点和元素这两个术语。但在必要时，我们将包括位置信息在内的存储块整体称为结点，而将其中的元素信息部分称为该结点的元素。

还可以有其他的存储数据的方法，如**索引**(index)的方法和**散列**(hash)的方法。这些基本

的存储表示方法，以及它们的组合，可用于实现数据的多种存储结构。

研究数据结构是为了解决应用问题，所以讨论数据结构必须同时讨论在数据结构上执行的相关运算及其算法才有意义。通过对运算及其算法的性能分析和讨论，使得我们在求解应用问题时，能选择和设计适当的数据结构，编写出高效的程序。

数据和操纵数据的运算是研究数据结构不可分割的两个方面，所以我们在讨论数据结构时，不但要讨论数据的逻辑结构，讨论数据的存储结构，还要讨论在数据结构上执行的**运算**(operation)，以及实现这些运算的**算法**(algorithm)。

1.2 数据抽象和抽象数据类型

抽象(abstraction)可以被理解为一种机制，其实质是抽取共同的和实质的东西，忽略非本质的细节。抽象可以使我们的求解问题过程以自顶向下的方式分步进行：首先考虑问题的最主要方面，然后再逐步细化，进一步考虑问题的某些细节，并最终实现之。

在程序设计中，抽象机制被用于两个方面：数据和过程。**数据抽象**(data abstraction)使程序设计者可以将数据元素间的逻辑关系与数据在计算机内的具体表示分别考虑。**过程抽象**(procedual abstraction)可使程序设计者将在数据上定义的运算与实现这些运算的具体方法分开考虑。抽象的好处在于降低了问题求解的难度。

封装(encapsulation)通常是指把数据和操纵数据的运算组合在一起的机制。使用者只能通过一组允许的运算访问其中的数据。从某种意义上说，数据的使用者只需知道这些运算的规范(定义)便可访问数据，而无需了解数据是如何组织和存储的，以及这些运算的具体算法如何等与实现有关的细节。也就是说对使用者隐藏了实现的细节。这种程序设计的策略称为**信息隐蔽**(information hiding)。

我们通常将数据和操纵数据的运算组成**模块**(module)，每个模块有一个明确定义的**接口**(interface)，模块内部信息只能经过这一接口被外部访问。一个模块的接口是实现运算的一组**函数**(functions)。这样的模块被称为**黑盒子**(blackbox)。一个程序使用一个采用信息隐蔽原则设计的模块被称为该模块的**客户**(client)。

封装和信息隐蔽有助于降低问题求解的复杂性，提高程序的可靠性。

读者已熟悉 C 语言的基本数据类型，它们包括字符型、整型、实型、指针类型等原子类型。原子数据类型的值是不可分割的。现实世界最终可以用这些初等数据来表示。另一类是结构类型。结构类型的数据的值是由若干成分按某种结构组成的，因此是可以分解的。

数组(array)和**结构**(structure)是 C 语言提供的两种组织数据的机制。

例如 int list[5]定义了 5 个整数的数组，其下标范围为 0 到 4。结构需要显式定义。例如：

```
struct student{
    int student_id;
    char name[20];
    char sex;
    int age;
};
```

数据类型是数据抽象的一种方式。一个**数据类型**(data type)定义了一个值的集合以及作用于该值集的运算的集合。程序设计语言中，数据类型不仅规定了该类型的变量(或常量)的取值范围，还定义了该类型允许的运算。例如一个类型为 int 的变量的取值范围是 -32 768～32 767，在整型数上的运算有+、-、*、/、%，关系运算<、>、<=、>=、==、!=，赋值运算=等。图 1-4 的上半部分给出了 int 的规范，它规定了整型变量(或常量)的取值范围及允许的运算；图的下半部分表示与该类型的实现有关的方面。两者是分离的。

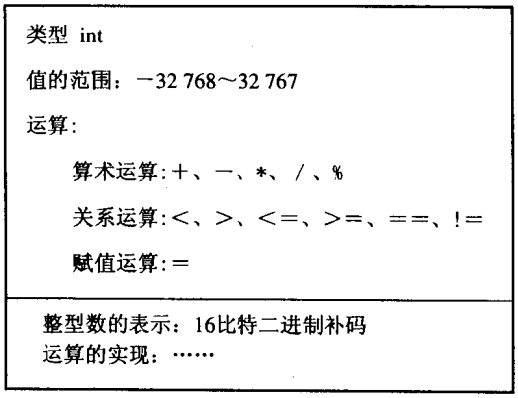


图 1-4 C 语言的数据类型 int

了解一个数据类型的数据对象(变量、常量)在计算机内的表示是有用的，但也是危险的。这使得应用程序可直接编写算法操纵该数据对象，但一旦改变该对象的存储表示，则必须改变所有使用该对象的程序。目前，普遍认为对使用者隐藏一个数据类型的对象的表示是个好的设计策略，即用户只能通过使用该类型提供的函数操纵该对象。这样，当数据对象的表示或实现函数的算法改变时，只要不改变函数的调用方式，应用程序将无需改变。

一个**抽象数据类型** ADT(abstract data type)是一个数据类型，其主要特征包括该类型的数据对象及其运算的规范，它与数据对象的表示和运算的实现分离，实行封装和信息隐蔽。在一个抽象数据类型上应当定义哪些运算，这取决于应用。若一组运算是完备的，那么我们可以使用该组运算，对该数据结构实行我们希望的所有操作。

其实，C 语言的类型 int 就是一个抽象数据类型，我们只能通过类型 int 所规定的运算操纵整型变量或常量。虽然仅在面向对象程序设计语言出现后，才提供了必要的机制，使用户可以实现抽象数据类型，如 C++语言的**类**(class)，但这并不意味着当我们使用 C 语言描述数据结构时，不能使用抽象数据类型的原则。

对于一个数据结构，一方面要说明它的数据的逻辑结构，同时还应定义一组在该数据结构上执行的运算。逻辑结构和运算的定义组成了数据结构的**规范**(specification)，而数据的存储结构和运算算法的描述构成了数据结构的**实现**(implementation)。规范是实现的准则和依据，规范指明“做什么”，而实现解决“怎样做”。从规范和实现两方面来讨论数据结构的方式是抽象数据类型的观点。在本书中，一种数据结构被作为一个抽象数据类型来加以讨论。

1.3 数据结构的描述

从 1.1 节中我们知道，在概念层次上，可以根据元素间的关系将数据结构划分为四种不同的结构：集合结构、线性结构、树形结构和图结构。其中，每一种结构又可进一步划分，得到多种数据结构。有时一种数据结构需借助其他数据结构来表示，比如集合结构可以使用线性表、搜索树或散列表来表示。

如上所述，一个数据结构被看成是一个抽象数据类型。一个数据结构的 ADT 描述的是 ADT 的接口，它由 ADT 名称，对数据的逻辑结构关系的简单陈述，以及该 ADT 上定义的一组运算的规范这几部分组成。

一个运算的规范是指在概念的层次上对一个运算的精确定义，或者说是从客户使用的角度描述的运算。它既能精确描述运算又不涉及运算的实现细节。本书中我们从语法和语义两方面描述一个运算。一方面我们使用 C 语言的**函数原型**(function prototype)规定该运算的使用格式，包括运算名称、运算的输入参数、输出参数和返回值，另一方面使用**前置条件**(precondition)和**后置条件**(postcondition)来定义一个运算。前置条件规定了使用一个运算应当满足的先决条件。后置条件规定了运算执行后应有的结果，即运算的作用。如果某次调用满足前置条件，则该运算的任何实现应保证运算执行后得到后置条件所规定的结果。运算的调用者有责任保证前置条件成立。这里需说明一点，运算是在数据结构较抽象的层次上的概念，当我们实际用 C 语言函数实现运算时，有时还需要辅助函数，也就是说实现一个运算可能需要不止一个 C 语言函数。

下面我们以复数数据结构 Complex 为例，介绍本书中我们所采用的描述数据结构的方法和格式。复数结构 Complex 被定义为一个抽象数据类型 ADT Complex。在 ADT 1－1 Complex 的描述中，我们使用了格式化的自然语言来描述。复数结构由一对实数(x，y)构成，x 为实部，y 为虚部。在复数上定义了构造函数(Comp)、加(Add)、减(Sub)、乘(Mul)和除(Div)四个运算。我们常常可以根据需要定义多于一个的构造函数。

ADT 1－1 Complex{

数据：

由一对实数(x，y)构成，x 为实部，y 为虚部.

运算：设两个复数分别为 a=(a_1，a_2)和 b=(b_1，b_2)。

Complex Comp(float x，float y)

后置条件：构造函数，函数返回复数(x，y)。

Complex Add(Complex a，Complex b)

前置条件：和的实部和虚部分别不超过实型值的允许范围。

后置条件：返回复数(a_1+b_1，a_2+b_2)。

Complex Sub(Complex a，Complex b)

前置条件：差的实部和虚部分别不超过实型值的允许范围。

后置条件：返回复数(a_1−b_1，a_2−b_2)。

Complex Mul(Complex a，Complex b)