21世纪高等院校计算机系列教材

Computer

□ 李克清 主编

数据结构

——C语言描述

华中科技大学出版社

21 世纪高等院校计算机系列教材

数据结构——C 语言描述

主　编：李克清

副主编：夏祥胜　崔洪芳

张万山　章　英　操保华

华中科技大学出版社

图书在版编目(CIP)数据

数据结构——C语言描述/李克清　主编
武汉:华中科技大学出版社,2005年2月
ISBN 7-5609-3340-8

Ⅰ.数…
Ⅱ.李…
Ⅲ.数据结构-教材
Ⅳ.TP311.2

数据结构——C语言描述　　李克清　主编

责任编辑:彭保林　曾　光　　封面设计:刘　卉
责任校对:陈　骏　　责任监印:熊庆玉

出版发行:华中科技大学出版社
武昌喻家山　　邮编:430074　　电话:(027)87557437

录　　排:北京搜获科技有限公司
印　　刷:荆州市今印印务有限公司

开本:787×1092　1/16　　印张:17.75　　字数:400 000
版次:2005年2月第1版　　印次:2005年2月第1次印刷　　定价:26.50元
ISBN 7-5609-3340-8/TP·555

内 容 简 介

本书根据“数据结构”课程教学大纲的要求，对常用的数据结构做了系统介绍，注重实际应用，概念清晰。全书共分九章，重点论述了数据结构的基本概念、线性表、栈和队列、递归、串、数组和广义表、树、图、查找、排序和文件。

本书叙述由浅入深、层次清楚、语言精练、逻辑性强，侧重于程序设计技术、算法和应用，各章中所涉及的数据结构与算法都给出了相应的C语言描述。本书主要面向本科院校的计算机类专业学生，也可以作为非计算机专业学生的选修课教材和其他计算机技术人员的参考书。

前　言

数据结构是计算机科学与技术专业教学计划中的一门核心课程，同时也是信息计算、电子信息技术等非计算机专业的一门重要专业基础课程。计算机科学与技术及其相关学科都会用到各种数据结构，数据结构已经成为计算机科学与技术工作者，尤其是计算机应用领域的开发人员的必备知识。

数据结构的任务是根据从各种实际问题中归纳、抽象出来的对象的数据特征和对象之间的关系，选择合适的数据组织方法、存储方法和相应的算法。这些方法有助于设计出周密、有效和风格良好的程序。

数据结构课程的教学要求是让学生学会分析和研究计算机加工的数据对象的特征，以便在实际应用中选择适当的数据逻辑结构、存储结构和相应算法，掌握算法的时间和空间性能分析技巧，学习应对复杂程序设计方面的技巧。

本书循序渐进地介绍了线性表、栈和队列、递归、串、数组和广义表、树、图、查找、排序和文件等知识。考虑到本书是一本面向计算机科学与技术专业和非计算机专业学生的教科书，本着简明、实用、通俗易懂的原则，尽量避免繁琐、深奥的理论推导和说明。因此，只要读者具备一定的C语言知识就能轻松地学习。

好的程序设计风格和方法不仅对计算机科学与技术专业的学生和软件工程者极其重要，对于非计算机专业的学生和应用程序设计来说也同等重要，因此全书采用C语言描述ADT，实现了各种数据类型的ADT说明、表示和实现。为了便于读者编程实践，本书给出的数据结构描述都是采用ANSI C风格的，读者补充相应的主函数(main)就可以在计算机上实现相应的数据结构。

本书计划学时是72学时，不同专业可根据自己的需要进行选择。值得指出的是，数据结构是一门实践性很强的课程，因此无论在校生还是自学者都应独立完成本书各章节的习题，并进行5次以上的上机实习。

全书共分为9章，其中第1、3、7、9章由李克清编写，第2章由张万山编写，第4章由章英编写，第5章由崔洪芳编写，第6章由操保华和王云合作编写，第8章由夏祥胜编写。全书由李克清任主编，康琼参加了后期的1、3、7、9章的教学课件制作。

由于作者水平有限，时间仓促，书中难免存在错误，恳请读者不吝指出，作者将不胜感激!

作　者

2004年12月

“21世纪高等院校计算机系列教材”丛书编委会

主任　何炎祥

委员（按姓氏拼音排序）：

戴光明	都志辉	桂　超	金先级
柯敏毅	李康顺	李克清	李禹生
刘腾红	卢强华	陆　迟	吕顺营
沈海波	石　清	王江晴	王伟军
王　忠	叶骏民	余敦辉	湛为芳

序　言

21世纪是信息时代，以计算机为核心的信息技术是21世纪科技发展的大趋势。作为计算机专业人才培养基地的大学计算机专业和相关专业，如何适应这种发展，培养出符合时代要求和社会欢迎的人才，是近年来计算机教育界讨论的热门话题，也是我们长期思考并努力探索的课题。

教材是人才培养的基础。在华中科技大学出版社的委托下，我们组织了有关高等院校的部分专家、教授共同编写了这套“面向21世纪计算机系列教材”，以期在适应21世纪的教材建设方面做出自己的努力。由于计算机行业发展日新月异，“21世纪计算机系列教材”编委会将负责系列教材的选题、每本教材大纲的编写和审定，以及教材、教学辅导书和课件的修订、更新等工作，以确保教材的正确性和先进性，使这套教材努力走在同类教材的前列。

这套系列教材包括计算机专业课和部分专业基础课教材，以及与之配套的实践课教材和教学辅导书等等。

我们希望这套教材具有以下特点：

1. 注重基础性和先进性的结合。计算机学科的一个显著特点就是知识和技术更新快，这对教学内容、课程知识结构的选取和组织提出了新的要求。我们把编写的重点放在基础知识、基本技能和基本方法上，希望在提高学生的理论素养和分析问题、解决问题的能力的同时，注重介绍新的技术和方法，以拓展学生的知识面，激发他们学习的积极性和创新意识。

2. 注重理论性与应用性的结合。良好的理论素养是应用的前提，而掌握理论的目的就是为了更好的应用。在教材的编写过程中，我们注意理论的系统性，在讲深讲透主要知识的基础上，融理论性和应用性于一体，注意基本方法的讲授，以培养学生应用理论和技术的能力。

3. 注重时代性和实用性的结合。力求精简旧的知识点，增加新的知识点，体现教材的时代特征。而且充分考虑一般高校目前所拥有的师资条件和教学设备，注重教材的实用性。

4. 注重科学性与通俗性的结合。概念、原理、新技术的阐述力求准确、精练；写作上尽量通俗易懂、深入浅出、图文并茂，增强可读性，便于学生自学。

5. 网络技术辅助教学。针对本系列教材我们开发有专门的网站（http://www.hzpress.org）、课件发布演示系统和考试系统等，以便为任课老师的教学提供更便捷、更全面的服务，并将通过网站开展各种形式的教材网上专家答疑、内容修订发布、课件定期升级等活动，以与读者随时互动，为读者提供立体化的服务。

教学改革是需要不断探索的课题。要达到以上目标，还需要不断地努力实践和完善。欢迎使用这套教材的教师、学生和其他读者提出宝贵意见。

最后，对参加这套教材编写的所有作者，对为这套教材的编写提供支持的有关学校、院系的领导和老师表示诚挚的谢意！感谢华中科技大学出版社为本系列教材的出版所付出的辛勤劳动！

教材编委会主任　何炎祥

（教授、博导、武汉大学计算机学院院长）

目　录

第1章 引 言

教学目标

- 了解数据结构的基本概念，数据结构的逻辑结构和存储结构之间的区别；
- 了解数据结构的ADT描述方法；
- 掌握结构化程序设计方法，特别是逐步求精和分而治之的设计方法；
- 深入理解算法的概念及其特点，学会用C语言描述算法；
- 了解算法的时间和空间复杂度。

信息技术的迅速发展，为计算机的发展提供了更为广阔的应用空间。计算机的应用领域不再局限于科学计算，已广泛应用于控制、管理和数据处理等非数值计算的工作中，与之相应地，计算机加工处理的对象也由纯粹的数值发展到字符、图、表和超文本等一些具有结构的数据对象，因此有必要研究这些带有结构的数据对象及其相关算法。

1.1 数 据 结 构

1.1.1 数据结构的概念

数据结构是专门研究计算机处理过程中经常遇到的各种典型问题的学科，研究问题涉及线性表、树、图和查找等，它的处理对象是已经抽象出来的各种数学模型。数据结构不仅要给出数据对象在计算机中的表示方式、存储形式，还要给出各种操作的算法实现。

研究数据结构时，关心的是数据对象的逻辑描述以及与数据对象相关函数的具体实现。数据对象的良好描述可以有效促进算法(在C语言中称为函数)的高效实现。

最常用的数据对象及函数已在C语言中被当作标准的数据类型加以实现，如整数对象(int)、实数对象(float)和布尔对象(bool)等。所有其他的数据对象均可以采用标准的数据类型、枚举(enum)、结构(struct)、联合(union)、数组(array)和指针(pointer)类型等所提供的组合功能来描述。例如，可以用string来描述长度为10个字符的字符串类型：

```
typedef char string[10];
```

【例1.1】学生成绩单如表1.1所示。从表1.1中可以看到，每个人的每门课成绩占一行。一个学生修完一门课，取得相应课程成绩后，学绩管理员就会在表中加上一行，因此成绩表会不断增加，而不会减少。当拿到成绩表时，您可能会查找自己或某个学生的全部成绩或某门课程的成绩；也可能会问“谁取得了数据结构课程的最高分？”“某位同学还有多少门课程没有通过考试？”“还要重修多少门课程？”等问题。其中的一些属性，如学生

姓名、班级、课程名称、开课学期、年度等都是字符串类型，而另一些属性，如成绩、学分则是数值类型的，这些属性一起组成学生成绩。为便于计算机处理学生成绩表，可以将这些属性整合起来组成一个结构(struct)，用以存放每个学生的各科成绩，然后把这些成绩按顺序存放在一起，形成叫做线性表的数据结构，如表1.1所示。

表1.1 学生成绩表

学生姓名	班级	课程名称	成绩	开课学期	年度	学分	…
马军	CS021	数据结构	92	春季	04	4	…
李娟	CS022	数据结构	88	春季	04	4	…
张萍	CS022	数据结构	78	春季	04	4	…
李娟	CS031	离散数学	85	秋季	04	4	…
…	…	…	…	…	…	…	…

【例1.2】 计算算法表达式(3 + 5) × 12 - sin(pow(2, 5))的值。

当从键盘上输入上述算术表达式后，要求计算机能给出正确的计算结果。为了能够正确地完成计算，必须要求计算机能够识别出上述表达式是一个加法(+)、减法(-)、乘法(×)、除法(/)或自定义函数表达式，从而识别输入串的计算类型，进而根据不同的计算类型做相应的处理。

上述表达式是减法表达式，因此表达式(3+5) × 12 - sin(pow(2, 5))被分成两个独立的子表达式(3+5) × 12和sin(pow(2, 5))，只要分别计算出(3+5) × 12和sin(pow(2, 5))的值，那么表达式(3+5) × 12 - sin(pow(2, 5))的值也就计算出来了。

对于子表达式(3+5) × 12和 sin(pow(2, 5))的计算，可以参照前面的计算过程分别独立完成，图1.1表示了表达式(3+5) × 12 - sin(pow(2, 5))的计算过程。

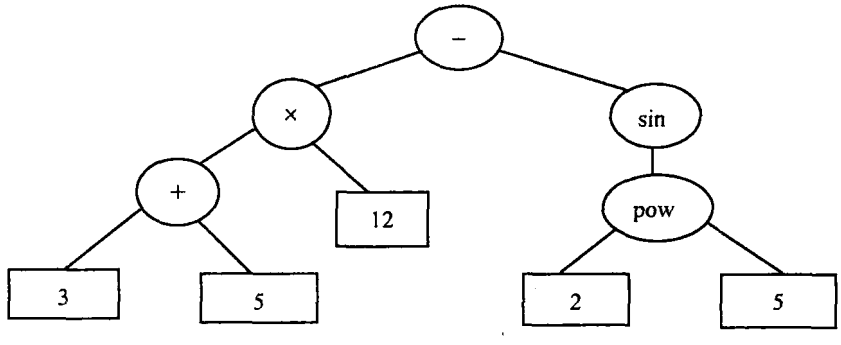

图1.1 表达式(3+5)×12−sin(pow(2,5))的计算过程示意图

计算顺序为先左后右，自底向上逐步进行。从图形上看，它和树的根结构相似。计算过程为：底层计算的结果传给上一层结点，直至最上面的根结点(root)，得到计算结果。图1.1表示的是一种被称作树的数据结构，形象地表示了计算结点之间的逻辑关系。

不难看出，例1.2中用到的数据结构(树)和例1.1中的数据结构(线性表)有很大区别，数据处理的方法也不尽相同。为了更好地研究这些具体问题，人们总结出使用计算机解决具体问题的基本步骤：首先从具体问题抽象出一个适当的数学模型，然后设计出一个解决此

数学模型的算法，最后经过编制程序、测试、调试程序直至得到该问题的解决方案[①]。寻求数学模型的实质是分析问题，从中提取操作的对象，并找出这些操作对象之间存在的关系，然后用数学语言加以描述。

为了讨论的方便，下面给出数据结构中涉及的一些基本术语。

数据(Data)是信息的载体。它能够被计算机识别、存储和加工处理，是计算机程序加工的“原料”。随着计算机应用领域的扩大，数据的范畴已经超越了简单的整数、实数、字符和字符串，而涵盖了图像、声音等复杂的对象。

数据元素(Data Element)是数据的基本单位。数据元素也称元素、结点、顶点或记录。一个数据元素可以由若干个数据项(Item，有些书上称之为字段、域、属性)组成，如例1.1表中的某一行信息就是一个数据元素，它含有学生姓名、班级等数据项。数据项是具有独立含义的最小标识单位，是数据不可分割的最小单位。

数据结构(Data Structure)是相互之间存在一种或多种特定关系(函数)的数据元素的集合。在这个定义中，强调的是“数据元素的集合”，体现了现代程序设计中以数据为中心的思想。可从以下三方面来理解数据结构的含义。

(1) 数据元素之间的逻辑关系，也称数据的逻辑结构(Logical Structure)。从逻辑关系上描述数据，独立于计算机且与数据的存储无关。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。

(2) 数据元素及其关系在计算机存储器内的表示，称为数据的存储结构(Storage Structure)。数据的存储结构是逻辑结构用计算机语言的实现(亦称为映像)，它依赖于计算机语言和计算机体系结构。对机器语言来说，存储结构是具体的。对所有的高级语言来说，存储结构是逻辑层面上的，且要通过编译程序才能转化为具体的存储结构。本书只在高级语言的层次上讨论存储结构。

(3) 数据的运算是对数据施加的操作。数据的运算定义在数据的逻辑结构上，每种逻辑结构都有一个运算的集合。最常用的有检索、插入、删除、更新和排序等运算。

数据的逻辑结构在形式上可以用二元组DS = (D，S)表示，其中D是结点的有穷集合，S是D上关系的有穷集合。

1.1.2 数据结构的分类

按照数据的逻辑结构，可以将数据结构分为两大类。

(1) 线性结构。若结构是非空集，有且仅有一个开始结点和一个终端结点，并且其他所有的结点只有一个直接前趋和一个直接后继。线性表是一个典型的线性结构，栈、队列和串等是线性结构的。

(2) 非线性结构。一个结点可以有多个直接前趋和直接后继。数组、广义表、树和图等数据结构是非线性结构的。

按照数据的存储结构，可以将数据结构划分为四大类。

(1) 顺序存储方法，把逻辑上相邻的结点存储在物理位置上相邻的存储单元里，结点

① 请参见1.3节的结构化程序设计。

间的逻辑关系由存储单元的邻接关系来体现。由此得到的存储表示称为顺序存储结构(Sequential Storage Structure)，通常借助程序语言的数组描述。该方法主要应用于线性的数据结构，非线性的数据结构也可通过某种线性化的方法实现顺序存储。

(2) 链接存储方法，不要求逻辑上相邻的结点在物理位置上也相邻。结点间的逻辑关系由附加的指针字段表示。由此得到的存储表示称为链式存储结构(Linked Storage Structure)，通常借助于程序语言的指针类型描述。

(3) 索引存储方法，在储存结点信息的同时，还建立附加的索引表。索引表由若干索引项组成。若每个结点在索引表中都有一个索引项，则该索引表称之为稠密索引(Dense Index)。若一组结点在索引表中只对应一个索引项，则该索引表称为稀疏索引(Spare Index)。

索引项的一般形式是：关键字、地址。关键字是能惟一标识一个结点的数据项。稠密索引中索引项的地址指示结点所在的存储位置；稀疏索引中索引项的地址指示一组结点的起始存储位置。

(4) 散列存储方法，根据结点的关键字直接计算出该结点的存储地址。

这四种基本存储方法，既可单独使用，也可组合起来对数据结构进行存储映像。同一逻辑结构采用不同的存储方法，可以得到不同的存储结构。选择何种存储结构来表示相应的逻辑结构，视具体要求而定，主要考虑运算方便及算法的时空要求。

1.2 抽象数据类型

抽象数据类型(Abstract Data Type，ADT)是指一个数学模型及定义在该模型上的一组操作(运算)。抽象数据类型的定义取决于它的一组逻辑特性，不涉及其具体实现的技术细节，因此不论问题的内部实现方式如何变化，只要它的数学模型没有改变，那么它的外部使用就不会受到任何影响。从这层意义上看，ADT是数据的逻辑结构及其在逻辑结构上定义的操作，而与其在计算机内部的表示和实现无关。

抽象数据类型不仅局限于计算机中已经定义并实现的数据类型(固有数据类型)，还包括用户设计软件系统时自行定义的数据类型。为了提高软件的复用率，近代程序设计方法学中指出：一个软件系统的框架应建立在数据之上，而不是建立在操作之上。即在构成软件系统的每个相互独立的模块上，定义一组数据和施于这些数据上的一组操作，并在模块内部给出这些数据的表示及其操作细节，而在模块外部使用的只是抽象的数据和抽象的操作。显然，所定义的数据类型的抽象层次越高，该抽象数据类型的软件模块的复用程度就越高。

ADT的定义可以用三元组<D，S，P>表示。其中D是数据对象，S是D上的关系集，P是对D的基本操作集。可以采用以下格式定义ADT：

```
ADT 抽象数据类型名 {
    数据对象：<数据对象的定义>
    数据关系：<数据关系的定义>
    基本操作：<基本操作的定义>
} ADT 抽象数据类型名
```