面向21世纪高等院校计算机教材系列

数据结构

张选平 雷咏梅 编
严蔚敏 审

机械工业出版社
China Machine Press

面向21世纪高等院校计算机教材系列

数 据 结 构

张选平　雷咏梅　编

严蔚敏　审

机械工业出版社

本书从数据类型角度系统地介绍了各种类型的数据结构的逻辑特性、存储表示及其基本操作算法，并针对常用的数据结构，进一步讨论各种应用算法及其实现方法。全书共分 10 章，采用 C 语言作为数据结构和算法的描述语言。第 1 章介绍数据结构和算法的有关概念，在第 2～6 章中，分别介绍了线性表、栈、队列、串、数组、树、图等主要数据结构及有关算法，第 7～10 章介绍集合类型数据结构的排序和查找方法，讨论了数据的组织结构和相应的排序和查找算法。

本书注重理论与实践相结合，每章配有适量例题和习题（包括上机实习题），以加强学生对相关内容的理解和应用，适用于教学和自学。

本书不仅可作为普通高等院校计算机类专业的教材，也可作为非计算机类专业学生的教材和教学参考书。

图书在版编目（CIP）数据

数据结构/张选平，雷咏梅编. —北京：机械工业出版社，2002.2
面向 21 世纪高等院校计算机教材系列
ISBN 7-111-09824-2
Ⅰ. 数… Ⅱ. ①张…②雷… Ⅲ. 数据结构—高等学校—教材 Ⅳ. TP311.12

中国版本图书馆 CIP 数据核字（2002）第 002331 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策　　划：胡毓坚
责任编辑：周艳娟
责任印制：路　琳

中国建筑工业出版社密云印刷厂印刷·新华书店北京发行所发行

2002 年 2 月第 1 版·第 1 次印刷

787mm×1092mm 1/16·18.75 印张·459 千字

0001－5000 册

定价：27.00 元

出 版 说 明

随着计算机技术的飞速发展，计算机在经济与社会发展中的地位日益重要。在高等院校的培养目标中，都将计算机知识与应用能力作为其重要的组成部分。为此，国家教育部根据高等院校非计算机专业的计算机人才培养目标，提出了“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次教育的课程体系。根据计算机科学发展迅速的学科特点，计算机教育应面向社会，面向潮流，与社会接轨，与时代同行。随着计算机硬件的不断更新换代，计算机教学内容也必须随之不断更新。

为满足高等院校计算机教材的需求，机械工业出版社聘请了清华大学、北方交通大学、北京邮电大学等高等院校的老师，经过反复研讨，结合当前计算机发展需要和编者长期从事计算机教学的经验，精心编写了“面向 21 世纪高等院校计算机教材”系列丛书。

本套教材理论教学和实践教学相结合，图文并茂、内容实用、层次分明、讲解清晰、系统全面，其中溶入了教师大量的教学经验，是各类高等院校、高等职业学校及相关院校的最佳教材，也可作为培训班和自学用书。

前　言

计算机技术的迅猛发展已远远超过了人们预料，它的应用范围迅速扩展，不再局限于科学计算，已渗透各个领域，如控制、管理等许多非数据处理领域。相关的数据处理对象也从数值发展到字符、表格和图形等带有结构的数据。因而在计算机中如何组织数据、处理数据就成了人们进行程序设计的关键所在。

数据结构就是研究数据以及数据之间关系的一门学科，研究内容主要包括：数据之间的逻辑关系、数据及其关系在计算机中的存储、对数据的运算或操作。计算机编程中加工处理的对象是数据，而数据具有一定的组织结构，所以学习计算机编程，仅仅了解计算机语言是不够的，还必须掌握数据组织、存储和运算的一般方法。因此，数据结构是从事计算机软件开发必备的专业知识，也是计算机专业一门非常重要的专业基础课。

本书是根据作者多年的教学经验，结合当前计算机科学技术的发展，特别是参考了 IEEE ACM 计算教程 2001 (2001 年 2 月发布)编著而成。内容覆盖了数据结构课程教学大纲的要求，从数据类型角度系统地介绍了各种类型的数据结构的逻辑特性、存储表示及其基本操作算法，并针对常用的数据结构，进一步讨论各种应用算法及其实现方法。目的在于培养学生分析研究计算机加工的数据对象的特征，以便选择适当的数据结构和存储结构以及相应的算法，并掌握算法时间复杂度和空间复杂度的基本技巧。

在内容组织上，本书以数据的逻辑结构为主线，分别介绍了线性结构、树型结构、图型结构以及集合类型的数据结构，全书共分 10 章。第 1 章综述数据、数据结构和抽象数据类型的基本概念，介绍算法分析和评价的基本思想；第 2～4 章介绍各种线性数据结构，其中第 2 章线性表是一种最典型的线性结构，第 3 章栈、队列和串是一些常用的特殊线性表，第 4 章数组和广义表是线性表的推广；第 5 章介绍树型结构，包括树和二叉树；第 6 章介绍图型结构；第 8～10 章介绍集合类型数据结构的排序和查找方法及数据的组织结构，其中第 7 章介绍各种排序方法，第 8 章介绍各种查找方法，第 9 章介绍数据的索引和散列组织结构，第 10 章介绍常用的文件组织结构和外部排序方法。

本书在介绍各种数据结构时，首先从数据的逻辑结构这个抽象层次出发，讨论数据对象的特性，再进一步介绍在计算机中的各种存储表示，在此基础上介绍各种操作算法的实现方法。全书采用 C 语言作为数据结构和算法的描述的工具，便于读者在计算机上实现。

本书可作为普通高等院校计算机类专业的教材，也可作为信息类相关专业学生的教材和教学参考书。本书对于从事计算机应用工作的科技工作者，也是一本实用的参考书。

清华大学严蔚敏教授在繁忙的教学科研工作中抽出时间对全书进行了审阅，在此谨表衷心的感谢。

由于作者水平所限，书中疏漏和错误之处在所难免，恳请广大同行和读者指正。

作　者

目　录

第1章 绪 论

计算机技术的迅猛发展已远远超过了人们预料，它的应用范围迅速扩展，不再局限于科学计算，已渗透到各个领域，如控制、管理等许多非数据处理领域。相关的数据处理对象也从数值发展到字符、表格和图形等带有结构的数据，这就给程序设计带来许多新的问题。因而在计算机中如何组织数据、处理数据就成了人们进行程序设计的关键所在。掌握数据在计算机中的各种组织和处理的方法，则是人们继续深入学习的基础。

信息的表示是计算机科学的基础。大多数计算机的主要目标不仅是完成计算，更重要的是存储和检索信息。从存储空间和运行时间的实现角度来看，这些程序必须组织信息，以支持高效的信息处理过程。因此，研究数据结构和算法可以有效地支持程序的实现。

数据结构的学习可以帮助人们更好地进行程序设计。数据结构选择的正确与否，会使你的程序效率相差甚远，同时还影响程序代码的可重用性。要实现数据的高效存储，就必须研究“数据结构”。

1.1 数据结构的基本概念

自然界中许多问题是可以用数学工具进行描述的，如造桥中的受力分析，可以通过数学方程来描述。但是还有很多问题无法用数学方程描述。

如读者到图书馆借书的过程，首先关心的是图书馆有否此书以及书在哪里存放。这一问题的解决必须根据图书目录进行查询。因此无法用数学方程来描述，必须考虑其他方法。

又如计算机对弈问题，其实质是计算机根据一定的策略进行选择判断。对弈的过程就是从一种格局派生出多种格局，这一过程的描述和解决就需要数据结构的知识。

从以上实例可以看出，这些问题的描述不是数学方程，而是书目表、对局树等数据结构。使用计算机来解决实际问题，大致需要以下三个步骤：

(1) 分析实际问题，从中抽象出一个适当的数学模型。

(2) 设计一个解决此数学模型的算法。

(3) 编程、调试、测试、修改直至最终获得满意解答。

数据结构这门课程就是要解决这三个步骤中的第一步和第二步中所提到的问题。数据结构是研究在非数值计算的程序设计问题中，计算机的操作对象以及它们之间的关系和操作的学科。实质上好的程序设计就是好的算法加之好的数据结构。

数据结构是编译原理、操作系统、数据库、信息检索和人工智能等课程的基础。作为一门课程，最早是由美国的唐·欧·克努特教授在 1968 年提出的，几十年来，数据结构技术已有了很大的发展提高，它是计算机专业的核心课程之一，是计算机学科的专业基础课。同时，也是非计算机专业的主要选修课程之一。

下面介绍一下数据结构的基本概念与术语。

数据(Data)就是指计算机能接受和处理的一切对象。

实际上，随着计算机硬件、软件和外部设备的不断发展，数据的含义越来越广泛。整数、实数、复数是数据，字符、文字、表格、图形、图像、声音等计算机也能够接受和处理，所以也是数据。

在计算机发展的初期，人们主要用它来进行数字计算，如处理一些线性代数、非线性代数、常微分方程、偏微分方程、极数、函数逼近等，因而人们当时把计算机称之为数据处理机。

据统计，计算机发展到现在，非数字计算已占 90%以上的计算时间。计算机加工处理的对象由纯粹的数值发展到目前的字符、表格、图形、图像、声音等具有一定结构关系的数据，因此，我们必须研究处理对象的特性及它们之间的关系。

数据元素(data element)是指在计算机中作为一个整体考虑和处理的对象，是组成数据的基本单位。如在学校的学籍管理信息系统中，学生、课程等就是数据元素。一般来说，数据元素是由若干数据项(data item)组成的。如学籍管理系统中，学生是数据元素，而学生是由学号、姓名、性别、年龄等数据项组成。

数据对象(data object)是指性质相同的数据元素的集合，是数据的子集。如：自然数数据对象是集合 N={1，2，3，…}，动物数据对象是集合 A={ox，pig，dog，cat，…}等。数据对象（data object ）即一组实例或值，例如：

- Boolean ={false, true }
- Digit = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- Letter = {A, B, C, …, Z, a, b, …, z}
- NaturalNumber = {0, 1, 2, …}
- Integer = {0, ±1, ±2, ±3, …}
- String = {a, b, …, aa, ab, ac, …}

Boolean，Digit，Letter，NaturalNumber，Integer 和 String 都是数据对象，true 和 false 是 Boolean 的实例，而 0, 1, …,9 都是 Digit 的实例。数据对象的每个实例要么是一个原语（primitive）或原子（atomic），要么是由其他数据对象的实例复合而成。

数据结构(Data Structure) 是研究数据与数据之间关系的一门学科，它包括三个方面：

(1) 数据元素之间的逻辑关系描述，称为数据的逻辑结构(logic structure)。

(2) 数据在计算机中的表示方式，称为数据的存储结构(storage structure)或称物理结构(physical structure)。

(3) 数据所要进行的运算，称之为数据的操作(operation)。

数据结构包括数据对象和实例以及构成实例的每个元素之间所存在的各种关系。这些关系可由相关的函数来实现。当我们研究数据结构时，关心的是数据对象（实际上是实例）的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地促进函数的高效实现。

在数据的逻辑结构中，我们所强调的是数据元素之间的逻辑关系。

例如：a_0，a_2，...，a_{n-1}是一种被称之为线性表的数据结构，a_0是表中的第一个数据元素，a_{n-1}是表中的最后一个数据元素，a_i是与 a_{i+1}直接相邻的前一个数据元素，a_i是与 a_{i-1}直接相邻的后一个数据元素，这就是线性表数据元素之间的逻辑关系。a_i被称之为结点(node)，结

点中的内容可以是任意的，如：

1）650，173，07 ，271，569。

2）Italy，Japan，America，China，German。

3）学生成绩表。

数据的逻辑结构是从逻辑上来观察数据和数据之间的关系，它与数据在计算机中的存储无关，是独立于计算机的。而数据的存储结构是逻辑结构在计算机存储器中的实现，是依赖于计算机的。所以，研究数据的存储结构就是研究数据和数据之间关系的计算机表示。

数据的逻辑结构按关系分为线性结构（关系是线性的）和非线性结构（关系是非线性的）。线性结构包括线性表（典型的线性结构。如表 1-1 的学生成绩登记表是一个线性表的例子）、栈和队列（特殊的线性表，是具有特殊限制的线性结构，特殊限制是指数据运算只能在表的一端或两端进行）、串（也是特殊的线性表，其特殊性表现在它的数据元素仅由一个字符组成）、数组（是线性表的推广，它的数据元素是一个线性表）、广义表（也是线性表的推广，它的数据元素是一个线性表，但不同构，即或者是单元素，或者是列表）。非线性结构包括树（具有多个分枝的层次结构）和二叉树（具有两个分枝的层次结构）、有向图（一种网状结构，边是顶点的有序对）和无向图（一种网状结构，边是顶点的无序对）以及集合类型。几种逻辑结构我们用一个层次图描述，如图 1-1 所示。

表 1-1　学生成绩登记表

学　号	姓　名	课　程 1	课　程 2	课　程 3
20000303	liu	89	99	70
20000304	wang	83	89	93
20000305	zhang	90	83	69
20000306	zhu	78	19	90

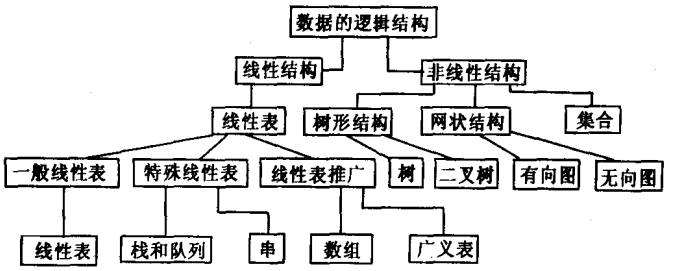

图 1-1　数据逻辑关系图

计算机的存储器是由许多个存储单元组成的，每个存储单元有唯一的地址。数据在计算机中的存放有四种基本的存储方法。

（1）顺序(Sequential)存储方法：就是把每个结点的数据，按某种顺序存放在一段连续的存储单元中。 然而顺序存储方法在下列情况下是无能为力的：其一是无足够大的连续存储空间，只有零碎小块；其二是事先无法得知所需存储空间的大小。

（2）链式(Linked)存储方法：就是把每个结点的数据，零散地存放在存储单元中。这

种方法就是将结点所占的存储单元分为两部分：一部分存放结点本身的信息，另一部分存放此结点的后继结点所对应的存储单元的地址，称为指针项。指针项也可以有多个。

（3）索引(Index)存储方法：就是用结点的索引号 i 来确定结点的存储地址，把每个结点的数据按一定规律顺序或链式存放在存储单元中。

（4）散列(Hash)存储方法：就是把每个结点的数据通过一个预设的散列函数，来决定该结点的存储单元。

数据的运算是定义在数据的逻辑结构上的一组操作，每种数据结构都有一个运算的集合。本课程主要研究的问题是数据的运算如何在计算机中实现。

在计算机科学中，数据是加工处理的主要对象，要使计算机对数据的处理速度加快，效率提高，就必须研究数据在计算机内的存储形式及数据与数据之间的组合结构。所以，人们在计算机中加工数据时，必须将散乱的数据集合转化成结点集合。

根据结点与结点之间的不同组合关系，分为线性关系（表）和非线性关系(树、图和集合)。所以说，数据结构主要研究的是结点与结点之间的关系，而不是结点本身。

1.2 抽象数据类型的表示与实现

从抽象数据类型(Abstract Data Type，简称 ADT)的概念出发，研究数据结构是近几年来计算机科学界出现的一种新倾向。利用 ADT 研究数据结构的优点在于信息隐蔽、模块化，说明的精确性、简单性、完整性以及实现的独立性。

我们在研究数据结构过程中，希望能以一种超越特定实现方式与特定语言的方式描述数据结构。研究数据结构的重点集中于抽象数据类型。抽象数据类型是一种研究数据结构的方式，它独立于具体的语言，以及某种具体的实现方式，此外抽象数据类型还提供了如面向对象程序设计等优点。

在高级程序设计语言中，编程所涉及的每个变量、常量或表达式，都有一个确定的数据类型(Data Type)。它直接或间接地规定了程序在执行期间变量或表达式可能的取值范围以及允许进行的操作。因此，数据类型是一个数据的值的集合和定义在这个值集合上的一组操作的总称。一般说来，数据类型可分为两类：一类是非结构的原子类型(Atomic Data Type)，其取值是不可分解的。如 C 语言中的标准类型(整数、实数、字符等类型)、枚举类型、指针类型等。另一类是结构类型(Structure Type)，其取值是由若干成分按某种结构组成的，因此它是可分解的，而且它的成分可以是结构的，也可以是非结构的，如 C 语言中结构的值是由若干分量组成的，而每个分量可以是某一个标准类型，也可以是数组等结构类型。

1．抽象数据类型概念

抽象数据类型和数据类型实质上是一个概念。如果说有区别的话，那就是 ADT 的范畴更广，它不再局限于系统已定义并实现的数据类型，还包含用户自己定义的数据类型。所以，ADT 是指一个数学模型以及定义在该模型上的一组操作。

根据这个定义可以得知，ADT 的定义仅取决于它的一组逻辑特性，与其在计算机内的表示和实现无关。因此，不论 ADT 的内部结构如何变化，只要其数学特性不变，都不影响其外部的使用。从而为实现软件的部件化和可重用性，提供了理论保证，进而提高了软件生

产率。

抽象数据类型的最重要的特点是抽象和信息隐蔽。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，从而使所设计的数据结构更具有一般性，可以解决一类问题。信息隐蔽就是对用户隐藏数据存储和操作实现的细节，使用者仅需了解抽象操作，或界面服务，通过界面中的服务来访问这些数据。

如图 1-2 所示用户与栈结构的关系。

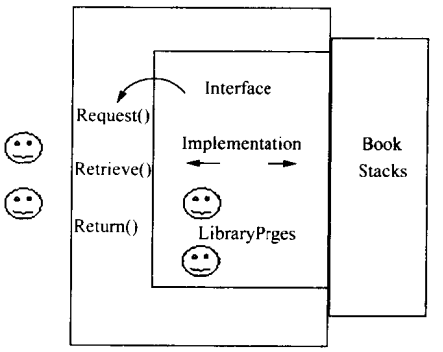

图 1-2　信息隐蔽示意图

Request（）：检查栈结构，回答读者所借图书是否存在；

Retriere（）：栈顶取书，送书给读者；

Return（）：放书入栈，将读者所还书放入书架。

【例 1-1】整数的数学概念和施加到整数的运算构成一个 ADT。C 语言的变量类型 int 就是对这个抽象数据类型的一种物理实现。

【例 1-2】一个整数线性表的 ADT 应包含下列操作：

插入一个新的整数到线性表的末尾；

按线性表元素的顺序依次打印整数；

如果线性表中存在一个特殊的整数就返回 TRUE，否则就返回 FALSE；

删除线性表中特定位置上的整数。

通过上述描述，每个操作的输入输出都清晰可见，但是线性表的实现还未做详细说明。

在软件模块中，一个抽象数据类型通常包含定义、表示、实现三个部分。其中，定义是公共属性，对其他编程者是可见的；表示和实现是私有属性，对其他编程者是不可见的。

抽象数据类型的定义是由一个值域和定义在这个值域上的一组操作构成的。而抽象数据类型的表示和实现则要依赖于所使用的语言功能。到目前为止，ADT 的表示和实现概括起来应该有这样三种情况：

第一种情况是面向过程的表示和实现，如标准 Pascal 语言，C 语言等。

第二种情况是面向模块结构的表示和实现，如 Turbo Pascal 4.0 版中的 Unit。

第三种情况是面向对象技术的表示和实现，如 Visual C++和 Visual Basic 等。

2. 抽象数据类型的说明

抽象数据类型可用以下三元组表示：(D,S,P)。

D 表示数据对象,S 是 D 上的关系，P 是对 D 的基本操作集。

ADT 抽象数据类型名{

数据对象：〈数据对象的定义〉

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

其中数据对象的数据关系的定义用伪代码描述，基本操作的定义格式为：

基本操作名（参数表）

初始条件：〈初始条件描述〉

操作结构：〈操作结构的描述〉

【例 1-3】构造三元组表。

```
ADT Triplet {
    对象：D={e1，e2，e3| e1，e2，e3∈ElemSet}
    关系：RI={〈e1，e2〉，〈e2，e3〉}
    操作：InitTriplet（&T，v1，v2，v3）
    操作结果：构造了三元组 T，元素 e1，e2，e3 分别被赋以参数 v1，v2，v3 的值。
    操作：Get(T ,i,&e)          取表中元素
}
```

1.3 算法设计与描述

为了更好地学习数据结构这门课程，本章主要介绍一下算法的基本概念、算法的描述形式及规则、算法的设计要求等有关知识。

1.3.1 算法

算法(Algorithm)是对特定问题求解步骤一种描述，是有穷的规则序列，这些规则决定了解决该特定问题的一系列运算，是指令的有限序列。在计算机中，算法是指令的有限集合，而每一条指令表示一个或多个操作。此外，算法还具有下列五个重要特性。

(1) 有穷性：就是指算法在执行一段时间后结束，不是无止境地执行下去。

(2) 确定性：算法中的每一条指令必须有明确的含义，人们理解不会产生二义性。并且在任何条件下，算法只有唯一的一条执行路径。也就是说，对于相同的输入，只能得出相同的输出。

(3) 可行性：就是指一个算法必须是能行的，即算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现的。

(4) 输入：一个算法有零个或多个输入，这些输入取决于特定的对象的集合。

(5) 输出：一个算法有零个或多个输出，这些输出与输入有某种特定的关系。

算法和程序是两个不同的概念。一个计算机程序被认为是对一个算法使用某种程序设计语言的具体实现。算法必须可终止意味着不是所有的计算机程序都是算法。操作系统是一个程序，而不是一个算法。然而，我们可以把操作系统的各种任务看成是单独的问题，每个问题由一部分操作系统程序通过特定的算法来实现，得到输出结果后便终止。

1.3.2 算法描述

一个算法可以用多种描述方法，大致的描述方法有如下几种：使用自然语言描述；使用计算机语言描述；使用形式语言描述等。

以下我们介绍一下 C 语言描述算法的一些方法。

算法的一般形式：

```
void  函数名(<参数表>)
{
<语句组>
}
```

函数

```
类型  函数名(<参数表>)
{
<语句组>
}
```

在参数表中，允许一切可能出现的变量类型。同时，除变量以外，所有函数中出现的局部变量都可不加类型说明。

1.4 算法的性能分析与度量

1.4.1 算法的性能标准

对算法设计的要求，在不同的场合，不同的应用领域，要求是不一样的，本书只讨论在数据结构课程中的一些要求。

(1) 正确性(correctness)：算法应当满足具体问题的要求。其正确性应满足如下要求，第一，算法对于几组输入数据能够得出满足规格说明要求的结果；第二，算法对于精心选择的典型、苛刻而带有特殊性的几组输入数据能够得出满足规格说明要求的结果；第三，算法对于一切合法的输入数据能够产生满足规格说明要求的结果。

(2) 可计算性（computability）：算法不能凭借直觉或经验进行描述。算法必须是每一步都严格定义的，必须能够一步一步精确执行。

(3) 可读性(readability)：算法是为了人的阅读与交流，其次才是机器执行。可读性好有助于人们对算法的理解，而不易读懂的程序，易于隐藏错误，难以调试和修改。

(4) 健壮性(robustness)：输入数据非法时，能做出反应，进行处理。一般方法是返回一个错误或值，终止程序，或更高抽象层次上进行处理。

(5) 通用性（generality）：我们所设计的算法具有一般性，即算法的运行结果对于一般数据集合都是成立的，而不是仅仅对于非常特殊的数据才成立。

(6) 效率及存储量要求：算法必须满足效率高，而且在给定的存储空间内正常运行。对同一问题的多个算法，尽量选用效率高（执行时间短）的算法。