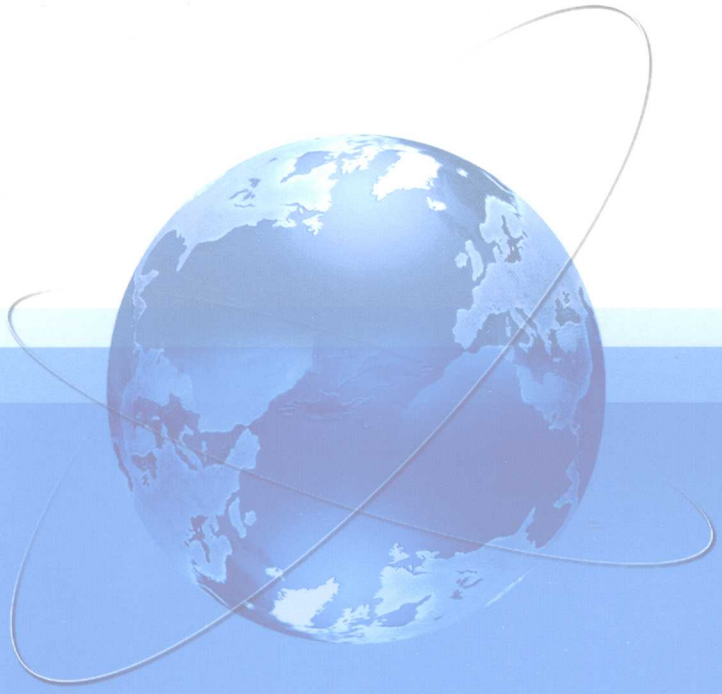
普通高等教育“十一五”国家级规划教材

21世纪高职高专规划教材（计算机类）

数据结构(C语言版)



吴子东　主编

配电子教案

机械工业出版社
CHINA MACHINE PRESS

普通高等教育 "十一五" 国家级规划教材
21世纪高职高专规划教材（计算机类）

数据结构

（C语言版）

主　编　天津大学职业技术教育学院　吴子东
副主编　西安理工大学高等技术学院　张爱玲
参　编　长治职业技术学院　郎贵义
浙江金华职业技术学院　楼建忠
大连职业技术学院　杜中一
张家界航空工业职业技术学院　刘伟跃
山东省高唐县职业教育中心　许　峰
天津大学职业技术教育学院　那一沙　许国栋
主　审　天津电子信息职业技术学院　田文成

机械工业出版社

本书是根据高职高专教育的特点、培养目标和教学要求而编写。全书共分8章，依次介绍了数据结构的基本概念、线性表、链接表、数组和广义表、树、图、查找和排序等。每章用大量的实例和图表来说明基本概念和方法。每章后配有丰富的练习题并给出了习题的参考答案。教材采用C语言与类C相接合的方式作为算法的描述语言，算法也尽可能地少用抽象定义，而更多的是给出具体算法，并力求算法更接近于实际应用，使读者能更快地提高编程能力。结合现代教育技术，教材配有多媒体课件以辅助教学，对抽象的数据结构辅之以形象的动画，不仅能提高学生的学习兴趣，也加深了对抽象概念的理解。

本书不仅适用于高职高专计算机类专业教学的需要，也可作为专业技术人员的参考用书。

本书配有电子教案，凡一次性购书30本以上者免费赠送一份电子教案。请与本书责任编辑余茂祚联系（联系电话010－88379759，邮箱yumaozuo@163.com）

图书在版编目（CIP）数据

数据结构：C语言版/吴子东主编．—北京：机械工业出版社，2008.2
普通高等教育“十一五”国家级规划教材．21世纪高职高专规划教材．计算机类
ISBN 978－7－111－23463－0

Ⅰ．数… Ⅱ．吴… 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．TP311.12 TP312

中国版本图书馆CIP数据核字（2008）第017961号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）
策划编辑：余茂祚 责任编辑：余茂祚 版式设计：
责任校对：李 婷 责任印制：邓 博
北京京丰印刷厂印刷
2008年4月第1版·第1次印刷
184mm×260mm·12.5印张·306千字
0 001—4 000册
标准书号：ISBN 978－7－111－23463－0
定价：21.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
销售服务热线电话：（010）68326294
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）68354423
封面无防伪标均为盗版

21世纪高职高专规划教材
编委会名单

编委会主任 王文斌

编委会副主任（按姓氏笔画为序）

王建明　王明耀　王胜利　王寅仓　王锡铭
刘　义　刘晶磷　刘锡奇　杜建根　李向东
李兴旺　李居参　李麟书　杨国祥　余党军
张建华　茆有柏　秦建华　唐汝元　谈向群
符宁平　蒋国良　薛世山　储克森

编委会委员（按姓氏笔画为序，黑体字为常务编委）

王若明　**田建敏**　成运花　曲昭仲　朱　强
刘　莹　刘学应　许　展　**严安云**　李连邺
李学锋　李选芒　**李超群**　**杨　飒**　**杨群祥**
杨翠明　吴　锐　何志祥　何宝文　余元冠
沈国良　张　波　**张　锋**　张福臣　陈月波
陈向平　陈江伟　武友德　林　钢　周国良
宗序炎　赵建武　恽达明　**俞庆生**　晏初宏
倪依纯　徐炳亭　**徐铮颖**　韩学军　崔　平
崔景茂　**焦　斌**

总　策　划　余茂祚

前　言

本教材为普通高等教育“十一五”国家级规划教材。数据结构是计算机专业中的重要课程之一，也是其他相关专业的主干课程。学生在初步掌握了计算机的基础知识和一种程序设计语言之后（本教材主要是指C语言），学习本课程可以明显地提高编程水平和解决实际问题的能力。本书采用C语言与类C相接合的方式作为算法的描述语言，这样既可以减少冗长的文字篇幅，又可以很容易将算法用C语言或其他语言来实现。教材在文字叙述上采用把理论性较强的部分尽量用通俗易懂的语言来描述，将一些繁杂的公式推导略去，而多增加一些带有总结归纳性的内容。教材在算法上也尽可能地少用抽象定义，而更多的是给出具体算法，并力求算法更接近于实际应用，这样可以使高职高专的学生能更快地提高编程能力。结合现代教育技术，教材配有多媒体课件以辅助教学，对抽象的数据结构辅之以形象的动画，不仅能提高学生的学习兴趣，也加深了对抽象概念的理解。

本教材共分8章。第1章主要介绍数据结构的基本概念和算法描述的基本思想。第2章讲解了线性表的定义、顺序存储结构和基本运算，还介绍了栈、队列和字符串。第3章介绍了线性表的链式存储结构和基本运算以及链栈、链队列和字符串的链式存储。第4章引入了数组和广义表的概念为后续课程打下了基础。第5章主要介绍树形结构的概念，在介绍树的基本概念后重点介绍二叉树的定义、性质、存储结构、各种遍历方法、一般树与二叉树的转换和哈夫曼树的应用。第6章主要介绍了图的基本概念、存储结构、图的遍历、求图的最小生成树、最短路径和图的拓扑排序。第7章讨论了查找的概念，主要介绍顺序查找、折半查找、索引查找、二叉排序树的查找、B-树的查找和散列查找。第8章讨论了排序的概念，主要介绍了插入排序、折半排序、希尔排序、快速排序、堆排序和归并排序等。教材基本上含盖了数据结构课程的主要内容，各院校可以根据实际情况决定内容的取舍。

为了保证本书的质量，各章至少有两人参加了撰写，多人参加了意见，先后多次通过现代通信技术E-mail、电话等保持着“热线”联系，以便集思广益，尽量使内容丰富而准确。参加编写的人员有吴子东、张爱玲、郎贵义、楼建忠、杜中一、刘伟跃、那一沙、许峰、许国栋。

吴子东副教授作为主编除了参加本书的编写工作外还主要承担了全书统稿和规范格式，数遍浏览全书、修改全书，同时参加了课件的部分制作与修改工作。各位参编人员对该书提出了不少宝贵意见和建议。各参编人员互相协作、严格认真、不怕返工和麻烦，付出了辛勤劳动。全书由天津电子信息职业技术学院田文成副教授主审，在审阅过程中田文成副教授几乎逐句斟酌，并提出了不少宝贵意见和建议。

此外，在本书编写过程中得到了天津大学职教学院和参编人员所在单位各级领导及同事们的帮助和支持。对此表示衷心的感谢。在写作过程中还参考了许多专业书籍和重要资料、文献，在此也一并向它们的作者表示衷心的感谢。

最后还需说明的是，限于作者的水平以及时间仓促，书中难免存在错误和纰漏，恳请专家、任课教师和广大读者批评指正，在此先致以谢意。

编　者

21世纪高职高专规划教材书目（机、电、建筑类）

高等数学(理工科用)(第2版)
高等数学学习指导书(理工科用)(第2版)
计算机应用基础(第2版)
应用文写作
应用文写作教程
经济法概论
法律基础
法律基础概论
C语言程序设计

工程制图(机械类用)(第2版)
工程制图习题集(机械类用)(第2版)
计算机辅助绘图—AutoCAD2005中文版
几何量精度设计与检测
公差配合与测量
工程力学
金属工艺学
机械设计基础
工业产品造型设计
液压与气压传动
电工与电子基础
电工电子技术(非电类专业用)
机械制造技术
机械制造基础
数控技术
专业英语(机械类用)
金工实习
数控机床及其使用维修
数控加工工艺及编程
机电控制技术
计算机辅助设计与制造
微机原理与接口技术
机电一体化系统设计
控制工程基础
机械设备控制技术
金属切削机床
机械制造工艺与夹具

冷冲模设计及制造
塑料模设计及制造(第2版)
模具CAD/CAM

汽车构造
汽车电器与电子设备
公路运输与安全
汽车检测与维修
汽车检测与维护技术
汽车空调
汽车营销学

工程制图(非机械类用)
工程制图习题集(非机械类用)
离散数学
电路基础
单片机原理与应用
电力拖动与控制
可编程序控制器及其应用(欧姆龙型)
可编程序控制器及其应用(三菱型)
工厂供电
微机原理与应用
模拟电子技术
数字电子技术
数字逻辑电路
办公自动化技术
现代检测技术与仪器仪表
传感器与检测技术
制冷原理与设备
制冷与空调装置自动控制技术
电视机原理与维修
自动控制原理与系统
电路与模拟电子技术
低频电子线路
电路分析基础
常用电子元器件

单片机原理及接口术术案例教程
多媒体技术及其应用
操作系统
数据结构
软件工程
微型计算机维护技术
汇编语言程序设计
VB6.0程序设计
VB6.0程序设计实训教程
Java程序设计
C++程序设计
Delphi程序设计
计算机网络技术
网络应用技术
网络数据库技术
网络操作系统
网络安全技术
网络营销
网络综合布线
网络工程实训教程
计算机图形学实用教程
动画设计与制作
管理信息系统
电工与电子实验
专业英语(电类用)

物流技术基础
物流仓储与配送
物流管理
物流运输管理与实务

建筑制图
建筑制图习题集
建筑力学(第2版)
建筑材料
建筑工程测量
钢筋混凝土结构及砌体结构
房屋建筑学
土力学及地基基础
建筑设备
建筑给排水
建筑电气
建筑施工
建筑工程概预算
房屋维修与预算
建筑装修装饰材料
建筑装修装饰构造
建筑装修装饰设计
楼宇智能化技术
钢结构
多层框架结构
建筑施工组织

目　录

第1章 绪 论

自从世界上的第一台电子计算机于1946年在美国面世以来，计算机的软硬件技术得到了飞速的发展。计算机的应用领域从最初的科学计算已发展到人类活动的各个领域。数据是计算机可以直接处理的最基本和最重要的对象。科学计算、数据处理、过程控制以及对文件的存储和检索及数据库技术等，都是对数据进行加工处理的过程。因此，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方法，并利用这些特性和关系设计出一个结构性好、效率高的算法和程序。数据结构这门学科就是在计算机的发展过程中应运而生的，它是计算机学科的核心课程，在计算机系统软件和应用软件中都要用到各种类型的数据结构。学好数据结构这门课程对于学习计算机专业的其他课程，如操作系统、编译原理（编译技术）、数据库管理系统、软件工程、人工智能等都是十分有益的。随着计算机的发展人们对数据结构越来越重视，认为程序设计的实质就是对要处理的问题选择一种好的数据结构，并在此结构上施加一种好的算法。著名的计算机科学家 N. Wirth 写的《算法 + 数据结构 = 程序》一书正是体现了这种观点。对数据结构这一学科的主要研究范围包括：数据的逻辑结构和物理结构，并在这种结构上定义相关的操作运算，设计并实现相应的算法，分析算法的效率，以达到提高程序质量的目的。其中，逻辑结构是指数据元素之间的关联方式；物理结构又称存储结构，是指在计算机中如何具体存储数据的各种逻辑结构。同一种逻辑结构可以有几种不同的物理结构来实现。因此，从某种意义上讲研究物理结构更为重要。

1.1 数据结构的基本概念和术语

1. 数据（Data） 数据是指能输入到计算机中并能被计算机处理的一切对象。在这里的所谓数据必须作广义的理解，它是指能够被计算机识别、存储和加工处理的一切信息。它可以是数值数据，也可以是非数值数据。数值数据是一些整数、实数等；非数值数据包括字符、文字、图形、图像、语音等。因此，大到一本书、一篇文章、一张图表、一幅照片、一首歌曲等，小到一条语句、一个单词、一个算式、一个数值、一个字符等都是数据。今后随着计算机的发展，数据的范围还将不断扩大。

2. 数据元素(Data Element) 数据元素是一个数据整体中相对独立的基本单位。由于数据的范围非常广泛,因此基本单位也是可大可小的。大到一本书、一张图表、一幅照片、一首歌曲等,小到一个数值、一个字符等都是数据元素。对于较大的单位,一个数据元素可由若干个数据项(Data Item)组成,例如,一本书的目录卡片就可以包括书名、作者、出版社、出版日期等数据项。

3. 数据对象（Data Object） 数据对象是具有相同性质的数据元素的集合，是数据的一个子集。在某个具体问题中，数据元素都具有相同的性质，属于同一数据对象，例如：整数集、实数集、字符集等，不论是有限集合还是无限集合，都是数据的一个子集。

4. 数据结构（Data Structure） 数据结构是指相互之间存在着一种或多种特定关系的数据元素的集合。在任何问题中，数据元素之间都不会是孤立的，在它们之间都存在着这样或那样的关系，这种数据元素之间的关系称为结构。数据元素之间的相互关系包括三个方面：

数据的逻辑结构、存储结构和运算方法。这三方面是相互联系又相互独立的。逻辑结构可以独立于存储结构，而存储结构确定了运算方法的具体实现。数据结构的组成见表 1-1。

表 1-1　数据结构的组成

逻辑结构	线性结构	线性表 队列 栈	每个数据元素有且仅有一个前驱元素（第一个元素除外），有且仅有一个后继元素（最后一个元素除外），元素之间是 1:1 的联系
	非线性结构	树形结构	每个数据元素有且仅有一个前驱元素（根结点元素除外），但可以有任意多个后继元素，数据元素之间是 1:N 的联系（N≥0）
		图形结构	每个元素可以有任意多个前驱元素和任意多个后继元素，数据元素之间是 M:N 的联系（N≥0，M≥0）
	集合结构		各数据元素属于同一个集合，数据元素之间是松散的联系
存储结构	顺序存储		是把逻辑上相邻的元素存储在物理位置相邻的存储单元中
	链式存储		对逻辑上相邻的元素不要求其物理位置一定相邻，元素间的逻辑关系可以通过指针的链接来表示
	索引存储		在存储数据元素本身之外，再建立一个指示逻辑数据与物理数据之间一一对应关系的表称索引表
	散列存储		由数据元素的值来确定存储地址，由于该地址是元素值的函数，因此，该函数也称散列函数
运算方法	加工型操作	插入	在原结构的某个位置上增添新的数据元素
		删除	在原结构的某个位置上删除数据元素
		更新	改变原结构某个位置上数据元素值，可以是插入和删除操作的组合
		排序	改变原结构中元素之间的逻辑顺序关系，使元素按升序或降序排列
	引用型操作	查找	寻找满足某些条件的数据元素的位置
		读取	取出满足某些条件的数据元素的内容

（1）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，它反映的是数据元素之间的相互关系。根据数据元素之间关系的不同特性，表 1-1 中的四种关系可以用图 1-1 来形象的表示。其中集合是数据元素之间关系极为松散的一种结构，因此也可用其他结构来表示它。

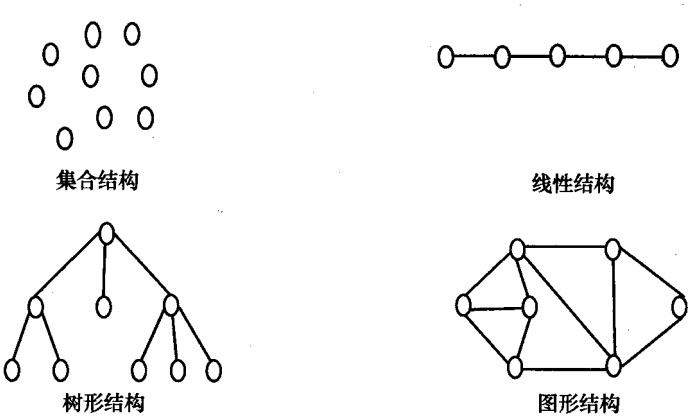


图 1-1　四种基本结构的示意图

为了更确切地描述逻辑结构，可以将逻辑结构用二元组表示：

Data _ Structure = (D,S)

式中 Data _ Structure——某一种逻辑结构；

D——该结构中数据元素的有限集合；

S——D 上关系的有限集合。下面举个简单例子说明之。

例 1-1 根据表 1-2 所给数据，用二元组表示法和图形表示法构造线性结构、树形结构和图形结构。

表 1-2 教务处人员简表

职 工 号	姓 名	性 别	出生年月	职 务	部 门
1	张国强	男	1960 年 8 月	处长	
2	李兰花	女	1969 年 6 月	科长	本科教务
3	马有年	男	1965 年 11 月	科长	高职教务
4	谭 力	男	1975 年 5 月	科长	培训
5	姜军旗	男	1954 年 3 月	科员	本科教务
6	赵 敏	女	1980 年 6 月	科员	本科教务
7	田淑霞	女	1970 年 8 月	科员	高职教务
8	王立军	男	1968 年 7 月	科员	高职教务
9	齐 越	男	1984 年 12 月	科员	培训

表 1-2 中共有 9 条记录，每条记录有 6 个数据项组成，每条记录的职工号是惟一的，所以可以把职工号作为该记录的关键字，在下面的例子中，用该关键字来代表整个记录。

1）线性结构的二元组表示：Line _ List = (D,S)，其中

D = {1,2,3,4,5,6,7,8,9}

S = { <5,1>, <1,3>, <3,8>, <8,2>, <2,7>, <7,4>, <4,6>, <6,9>}

对应的图形如图 1-2 所示。

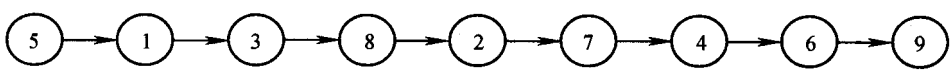


图 1-2 线性结构

从图 1-2 不难看出，S 是按职工年龄从大到小排列的关系。

2）树形结构的二元组表示：Tree = (D, S) 其中

D = {1,2,3,4,5,6,7,8,9}

S = { <1,2>, <1,3>, <1,4>, <2,5>, <2,6>, <3,7>, <3,8>, <4,9>}

对应的图形如图 1-3 所示。

从图 1-3 不难看出，S 是人员之间领导与被领导的关系。

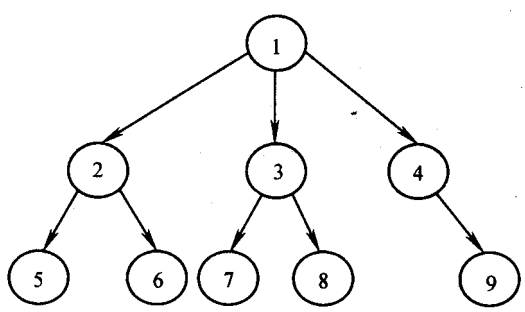


图 1-3 树形结构

3）图形结构的二元组表示：Graph = (D, S)，其中

D = {1,2,3,4,5,6,7}

S = { <1,2>, <2,1>, <1,4>, <4,1>, <2,3>, <3,2>, <2,6>, <6,2>, <2,7>, <7,2>, <3,7>, <7,3>, <4,6>, <6,4>, <5,7>, <7,5> }

对应的图形如图 1-4 所示。

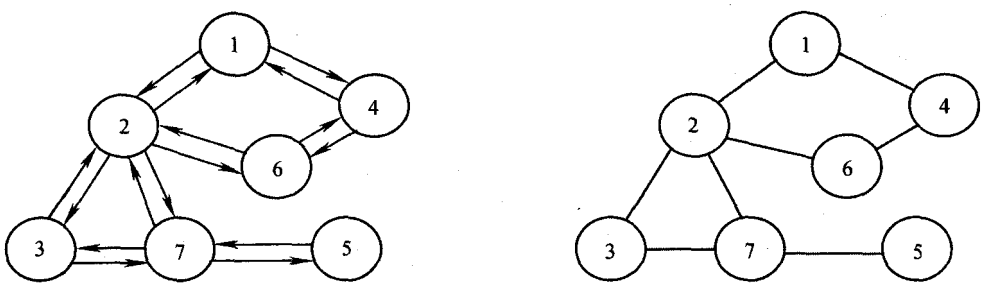


图 1-4　图形结构

从图 1-4 可以看出 S 是 D 上的对称关系，可以将这种对称关系简化，把 <x，y>，<y，x> 这两个序偶用一个（x，y）来代替；在图形中我们把 x 结点和 y 结点之间两条相反的弧用一条无向边来代替。这样 S 关系可以改写为

S = {(1,2),(1,4),(2,3),(2,6),(2,7),(3,7),(4,6),(5,7)}

对应的图形如图 1-4 右图所示。若 S 中每个序偶里的两个元素所代表的人员是好友的话，那么 S 关系就是人员之间的友好往来关系。

（2）物理结构（存储结构）。存储结构表示的是数据元素及其关系在计算机的存储器中的物理位置。由于已建立的逻辑结构是设计人员根据需求所选定的数据组织形式，因此，存储结构要遵循选定的逻辑结构。通常存储结构有四种基本方式，见表 1-1。每一种存储方式的应用在后续课程中讲述。

（3）运算方法。运算方法是对已建立的逻辑结构和存储结构的具体操作。运算与逻辑结构和存储结构是紧密地联系在一起的，在不同的结构上可以进行的运算是不同的。一般来说，根据操作的效果，可以将运算分为以下两类运算：① 加工型运算，其操作改变了原结构中数据元素的个数或数据元素的内容。② 引用型运算，其操作不改变原结构中数据元素的个数或数据元素的内容，只是从结构中提取某些信息作为运算的数据。两类运算所包含的基本运算见表 1-1。

5. 数据类型（Data Type）　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。它是和数据结构密切相关的一个概念。它最早出现在高级程序设计语言中，用以描述程序中操作对象的特性。可以理解为数据类型中定义了两个集合：值的集合和操作集合。其中值的集合定义了该类型数据元素的取值范围，操作集合定义了该类型数据允许参加的运算。在用高级语言编写的程序中，每个变量、常量或表达式都有一个它所属的确定的数据类型。类型明显地或隐含地规定了在程序执行期间变量或表达式所有可能的取值范围，以及在这些值上允许进行的操作。

按数据元素取值的不同特性，在高级程序设计语言中，数据类型可分为两类：一类是原子类型，另一类则是结构类型。原子类型的值是不可分解的。如 C 语言中整型、字符型、

浮点型、双精度型等基本类型。而结构类型的值是由若干成分按某种结构组成的，因此是可分解的，并且它的成分可以是非结构的，也可以是结构的。例如，数组的值由若干分量组成，每个分量可以是整数，也可以是数组等。在某种意义上，数据结构可以看成是“一组具有相同结构的值”，而结构类型可看成是由一种数据结构和定义在其上的一组操作所组成的。

6. 抽象数据类型（Abstruct Data Type，简称 ADT） 抽象数据类型是由一组数据结构和在该组数据结构上的一组操作所组成。抽象数据类型也可以定义为由一组数据和在该组数据上的一组操作所组成。也可以指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关。即在定义抽象数据类型中的数据部分和操作部分时，要求只定义到数据的逻辑结构和操作说明，不考虑数据的存储结构和操作的具体实现（即具体操作代码），也就是说不论其内部结构如何变化，只要它的数学特性不变，都不影响其外部的使用。

抽象数据类型和数据类型实质上是同一个概念。例如，各种计算机都拥有的整数类型就是一个抽象数据类型，尽管它们在不同处理器上的实现方法可以不同，但由于其定义的数学特性相同，在用户看来都是相同的。因此，“抽象”的意义在于数据类型的数学抽象特性。

抽象数据类型的定义可以由一种数据结构和定义在其上的一组操作组成，而数据结构又包括数据元素及元素间的关系，因此抽象数据类型一般可以由元素、关系及操作三种要素来定义。和数据结构的形式定义相对应，抽象数据类型可以用以下三元组表示：

$$(D, S, P)$$

式中 D——数据对象；

S——D 上的关系集；

P——对 D 的基本操作集。

抽象数据类型的特征是使用与实现相分离，实行封装和信息隐蔽。就是说，在抽象数据类型设计时，把类型的定义与其实现分离开来。因此，抽象数据类型概念的引入，软件设计的复杂性，使软件设计中普遍遵循的模块化、信息隐蔽、代码共享等思想得到充分体现。

7. 算法（Algorithm） 算法是对特定问题求解步骤的一种描述，是指令的有限序列。其中每一条指令表示一个或多个操作。一般算法都具有以下的特性：

（1）有穷性。一个算法必须在执行有限步之后结束，即必须在一定时间内完成。

（2）确定性。算法中的每一条指令必须有确切的定义，无二义性。在任何条件下，算法只有惟一的一条执行路径，即对于相同的输入仅有惟一的一条输出路径。

（3）可行性。可行性是指算法中描述的各种操作都可以通过已知的一组基本运算来实现。

（4）输入。一个算法可以具有零个或多个输入，这些输入取自特定的数据对象集合。

（5）输出。一个算法必须有一个或多个输出，这些输出是算法对输入进行运算的结果。

算法的含义与程序十分相似，但又有区别。程序中的指令必须是机器可执行的，而算法中的指令则无此限制。算法代表了对问题的求解，而程序则是算法在计算机上的特定的实现。一个算法若用程序设计语言来描述，则它就是一个程序。

算法与数据结构是相辅相承的。解决某一特定类型问题的算法可以选定不同的数据结构，而且选择恰当与否将直接影响算法的效率。

1.2 算法描述和算法分析

算法与数据结构的关系紧密，在算法设计时先要确定相应的数据结构，而在讨论某一种数据结构时也必然会涉及相应的算法。

1.2.1 算法描述

算法可以使用各种不同的方法来描述。

1. 自然语言　用自然语言来描述算法的优点是简单直观且便于人们对算法的阅读。缺点是不够严谨，与计算机的具体高级程序设计语言形式相差很大，需要用户进行转换。

2. 数学公式　用数学公式表达算法特别适应于数值计算的问题，最大优点是表达严谨。不足是局限性比较严重，和自然语言一样与计算机的具体高级程序设计语言形式相差很大。

3. 流程图　可以使用程序流程图（见图 1-5 右图）、N—S 图（见图 1-5 左图）等算法描述工具。其特点是描述过程简洁、明了。用以上两种方法描述的算法虽不能够直接在计算机上执行，但还是能比较容易地将它们转换成某种可执行的程序设计语言。

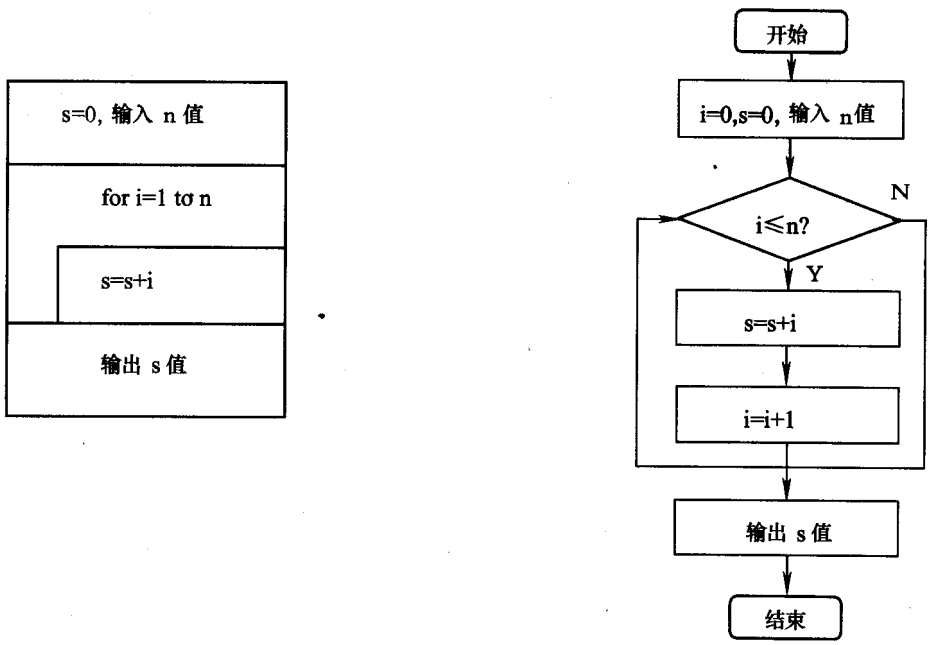


图 1-5　N—S 结构图和流程图

4. 程序设计语言和伪代码　可以直接使用某种程序设计语言来描述算法，不过直接使用程序设计语言并不容易，而且一些程序设计语言的语法要求非常严格使用并不方便，常常需要借助于注释才能看明白。

为了解决理解与执行两者之间的矛盾，人们常常使用一种称为伪码语言的描述方法来进行算法描述。伪码语言介于高级程序设计语言和自然语言之间，它忽略高级程序设计语言中一些严格的语法规则与描述细节，因此它比程序设计语言更容易描述和被人理解，而比自然语言更接近程序设计语言。它虽然不能直接执行但很容易被转换成高级语言。

5. C 语言作为算法描述语言　本教材采用 C 语言进行结构定义和算法描述，对于熟悉 C 语言的读者可以跳过本节。

（1）C 数据类型概述。C 语言的数据类型见表 1-3。

表 1-3　C 语言的数据类型

数据类型			变量类型说明符
基本类型	整型	短整型	int 或 short（signed short int）
		长整型	long（signed long int）
	实型	单精度	float
		双精度	double
		长双精度	long double
	字符型		char
构造类型	数组型		存储类型 数据类型 数组名［常量表达式］［常量表达式］…［常量表达式］;
	结构体型		struct 结构体类型名｛数据类型 成员名 1; …; 数据类型 成员名 n;｝
	联合体型		union　类型名｛数据类型 成员名 1; …;｝; union　类型名 联合体变量表;
	枚举型		enum 枚举类型名｛枚举常量名｝; enum 枚举类型名 枚举变量名;
指针型			存储类型 数据类型 *指针变量名; struct 结构体类型名 *结构体指针名;
空类型			void
用户自定义类型			typedef 类型名 1 类型名 2;

（2）C 语言使用的基本语句。C 语言的基本语句见表 1-4。

表 1-4　C 语言的基本语句

语句基本格式		基本功能
赋值语句	变量名 = 表达式	将表达式的值赋给变量。可以使用的赋值号有： =, +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=
	表达式语句	当一个表达式单独书写，并用分号结尾则称表达式语句，例：++i;或 i++;等
条件语句	if(表达式) 语句;	当表达式的结果为真（非 0）时执行后面的语句，否则不执行
	if(表达式)语句 1;else 语句 2;	当表达式 1 的结果为真时（非 0），则执行语句 1，否则执行语句 2
	if(表达式 1)语句 1;else if(表达式 2)语句 2;…else 语句 n;	分别计算并判断每一个表达式的值，若某个表达式的值为非 0，则执行其后的语句，执行完后退出该结构，若所有表达式的值均为 0，则执行语句为 n
开关语句	switch(表达式) ｛case 值 1:语句组 1;break; case 值 n:语句组 n;break; default 语句组 n+1;｝	执行此语句先计算表达式的值，然后用此值与 case 后面的值相比较，若与某个值相等，则执行后面的语句组。执行完后若有 break 语句，则退出此结构，若无 break 语句则继续执行后继语句（不再进行判断）。若所有值均不相等，但有 default 语句时，则执行语句组为 n+1
循环语句	while(表达式) 循环体语句;	先计算并判断表达式的值，若结果为真（非 0）时执行循环体语句，然后再计算并判断表达式的值……重复此过程直到表达式的值为假（0），结束循环
	do｛循环体语句 ｝while（表达式）;	先执行循环体语句，然后再计算并判断表达式的值，若结果为真（非 0）再执行循环体语句，直到表达式的值为假（0），退出循环
	for(表达式 1;表达式 2;表达式 3) 循环体语句;	先计算表达式 1 的值，再计算并判断表达式 2 的值，若结果为真（非 0）则执行循环体语句，执行完后计算表达式 3 的值，再计算并判断表达式 2 的值为真，执行循环体语句……直到表达式 2 的值为假（0）退出循环

（续）

语句基本格式		基本功能
函数调用	变量名 = 函数名(参数表)	以表达式的方式调用函数,并将函数的返回值赋给变量名
	函数名(参数表)	以语句的方式调用函数
	函数名(函数名(参数表1),函数名(参数表2)……函数名(参数表3))	以参数的方式调用函数,也可以将函数的返回值赋给变量名

(3) 常用动态存储函数。指针是实现动态存储结构的重要手段，在链表、树和图等动态存储结构的实现中，都需要使用指针，并经常使用C语言的动态存储分配函数。表1-5列出了几个常用的动态存储函数。

表1-5 常用的动态存储函数

函数名	函数原型	功能	返回值
calloc	void * calloc(unsigned n, unsign size)	分配n个数据项的内存连续空间，每个数据项的大小为size	返回分配内存单元的起始地址,若不成功返回0
free	void free(void * p)	释放p所指的内存区	无
malloc	void * malloc(unsigned size)	分配size字节的存储区	返回所分配的内存区起始地址,若内存不够返回0
realloc	void * realloc(void * p, unsigned size)	将p所指出的已分配内存区的大小改为size,size可以比原来分配的空间大或小	返回指向该内存区的指针

注意：在C语言中，数组的下标是从0开始的，为了便于分析，本教材规定定义的下标从0开始，但存放元素的下标从1开始。

1.2.2 算法分析

1. 算法设计要求　要设计一个好的算法通常要考虑以下的要求。

(1) 正确性。算法的执行结果应当满足预先规定的功能和性能要求。一个不能保证正确的算法，根本不能称为算法。

(2) 可读性。一个算法应当思路清晰、层次分明、简单明了、易读易懂。

(3) 健壮性。当输入不合法数据时，算法能适当作出反应或进行处理，不会产生莫名其妙的运行结果或死机。

(4) 高效率。一般指两方面。① 存储空间：在算法执行过程中所占空间越小越好。②时间效率：算法执行过程中执行的时间短的算法其效率较高。

在这些指标中除了正确性外,其余三方面往往是相互矛盾的。如当追求较短的运行时间时,可能会带来占用较多的存储空间和较复杂的算法;当追求较少的存储空间时,可能会带来较长的运行时间和较复杂的算法;当追求算法的简单性时,可能会带来占用较多的存储空间和较长的运行时间。而且随着计算机的飞速发展(指计算机的存储空间越来越大,运算速度越来越快),在实际评价中应根据需要有所侧重。数据结构课程着重讨论算法的时空性能。这并不意味着这一指标比别的指标更重要(实际情况往往正好相反),而仅仅是由于学习阶段的要求。

2. 算法性能分析与度量　可以从一个算法的时间复杂度与空间复杂度来评价算法的优劣。

(1) 时间复杂度。一个程序的时间复杂度是指程序运行从开始到结束所需要的时间。显然，在一个算法中，进行简单操作的次数越少，其运行时间也就相对地越少；次数越多，其运行时间也就相对地越多。若解决一个问题的规模为n。那么算法的时间复杂度就是n的一

个函数，通常记为 T（n）。下面用求累加和的算法介绍时间复杂度的计算。

```
float sum(int n)
{int i=1;
  float s=0,x;
  while (i<=n)                    n+1 次
  {scanf("%f",&x);                n 次
   s=s+x;                         n 次
   i++;                           n 次
  }
  return (s);                     1 次
}
```

把每一条指令执行后计算出的执行次数加起来，即得到时间复杂度 $T(n)=4n+2$。

但在许多时候要精确地计算 T(n) 是困难的，实际上，一般也没有必要精确地计算出算法的时间复杂度，只要大致计算出相应的数量级即可达到分析算法的目的。可以记作

$$T(n)=O(f(n))$$

式中 f(n)——问题规模为 n 的某个函数；

O——英文 Order（即数量级）一词的第一个字母。

这样表示的意思是指 T(n) 不超过一个正的常数同 f(n) 的乘积。算法的时间复杂度采用数量级的形式表示后，将给求一个算法的 T(n) 带来很大方便，这时只需要分析影响一个算法运行时间的主要部分即可，不必对每一步都进行详细的分析；同时，对主要部分的分析也可简化，一般只要分析清楚循环体内简单操作的执行次数即可。在上例求累加和的时间复杂度可以表示为

$$T(n)=O(n) \text{ 或记为 } O(n)$$

而 $T(n)=2n^2+3n+5$，可以记为

$$T(n)=O(n^2) \text{ 或记为 } O(n^2)$$

表 1-6 给出了各种有代表性的 T(n) 函数的算法在不同 n 值时的运行时间，表中凡未注明的时间单位为微秒（10^{-6}s）。因算法的运行时间随机器而异，所以此表上的时间主要用来作相对比较。

表 1-6　算法的运行时间与 T(n) 的关系

T(n)	$\log_2 n$	n	$n\log_2 n$	n^2	n^3	n^5	2^n	n!
n=4	2	4	8	16	64	1024	16	24
n=10	3.32	10	33.2	100	1ns	0.1s	1024	3.6288s
n=20	4.3	20	86.4	400	8ns	3.2s	1.05s	771 世纪
n=40	5.3	40	213	1600	64ns	1.7min	12.7d	2.59×10^{32} 世纪
n=60	5.9	60	354	3600	216ns	13min	366 世纪	2.64×10^{66} 世纪

从表中可以看出随着 n 值的增大，各种 T(n) 函数所对应的运行时间的增长速度大不相同，对数函数的增长速度最慢，线性函数较之快些，其余类推；因此，当 n 足够大后，各种不同数量级的 T(n) 函数存在着下列关系：

$$O(1)<O(\log_2 n)<O(n)<O(n\log_2 n)<O(n^2)<O(n^3)<\cdots<O(2^n)<O(n!)$$

通常用 O(1) 表示常数计算时间。使用大 O 记号表示的算法的时间复杂度，也称为算法的渐进时间复杂度（Asymptotic Complexity）。

（2）空间复杂度。一个程序的空间复杂度（Space complexity）是指程序运行从开始到结束所需的存储量，记作

$$S(n)=O(f(n))$$

其中 n 为问题的规模。一般情况下，一个程序在计算机上执行时，除了需要存储本身所用的指令、常数和输入数据以外，还需要一些对数据进行操作的辅助存储空间。程序运行所需的存储空间包括以下两部分：①固定部分：这部分空间与所处理数据的大小和个数无关，或者称与问题的实例的特征无关。主要包括程序代码、常量、简单变量、定长成分的结构变量所占的空间。②可变部分：这部分空间大小与算法在某次执行中处理的特定数据的大小和规模有关。例如对于用递归算法解决问题时，算法本身都比较短，占用的存储空间较少（固定部分）。但运行时需要一个附加堆栈，从而占用较多的临时工作单元（可变部分）；若写成非递归算法，算法本身都比较长，占用的存储空间较多（固定部分）。但运行时只需要较少的临时存储单元（可变部分）。

算法的时间复杂度和空间复杂度是相互矛盾的，难以兼得，即算法执行时间上的节约一定是以增加空间存储为代价的，反之亦然。

习　题

1. 选择题（把正确答案填在横线上）

（1）在数据结构中，从逻辑上可以把数据结构分成_____。

A. 动态结构和静态结构　　B. 紧凑结构和非紧凑静态结构

C. 线性结构和非线性结构　　D. 内部结构和外部结构

（2）执行下面程序段时，执行 S 语句的次数为_____。

```
for( i=1;i<=n;i++)
    for( j=1;j<=i;j++) S;
```

A. n^2　　B. $n^2/2$　　C. $n(n+1)$　　D. $n(n+1)/2$

（3）数据结构被定义为的 data _ structure =（D，S），其中 D 是①_____的有限集，S 是 D 上的②_____有限集。

①A. 算法　　B. 数据元素　　C. 数据操作　　D. 逻辑关系

②A. 操作　　B. 映像　　C. 存储　　D. 关系

（4）下面程序段的时间复杂度为_____。

```
for( i=1;i<=m;i++)
  for( j=1;j<=n;j++)
    a[i][j]=i*j;
```

A. $O(m^2)$　　B. $O(n^2)$　　C. $O(m*n)$　　D. $O(n+m)$

（5）计算机算法指的是①_____，它必须具备输入、输出和②_____等五个特性。

①A. 计算方法　　B. 排序方法

C. 解决问题的有限运算序列　　D. 调度方法

②A. 可执行性、可移植性和可扩充性　　B. 可行性、确定性和有穷性

C. 确定性、有穷性和稳定性　　D. 易读性、稳定性和安全性

2. 填空题

（1）数据的逻辑结构被分为_____、_____、_____和_____四种。