

现代高职高专信息技术教材

数据结构

林志英 付百文 朱立平 编著
鲍有文 主审

中国高等职业技术教育研究会
全国高等学校计算机基础教育研究会高职高专专业委员会
组编 推荐

中国林业出版社
China Forestry Publishing House
www.cfph.com.cn



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

现代高职高专信息技术教材

数据结构

林志英 付百文 朱立平 编著
鲍有文 主审

中国高等职业技术教育研究会
全国高等学校计算机基础教育研究会高职高专专业委员会
组编推荐

中国林业出版社
China Forestry Publishing House
www.cfph.com.cn



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

本书由 10 章构成, 内容包括数据结构的概念、算法及算法分析, 线性表、栈和队列的概念及应用, 串、多维数组和广义表, 二叉树、哈夫曼树、森林、图及其应用, 排序方法以及文件等。

本书内容安排合理、重点突出、讲解清楚, 突出面向实践、重在应用的特点, 为了便于教学, 书中配有相当数量的例题和一定数量的习题, 并配有综合练习题及实验, 主要让学生掌握数据结构的基本概念和各种算法实现。

本书可作为高职高专及各类高等院校计算机相关专业数据结构课程的教材, 也可作为非计算机专业相关选修课教材。

图书在版编目 (CIP) 数据

数据结构/林志英 付百文 朱立平编著. —北京: 中国林业出版社: 北京希望电子出版社, 2006.12
现代高职高专信息技术教材
ISBN 7-5038-4534-1

I. 数... II. ①林...②付...③朱... 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 (2006) 第 083172 号

出版: 中国林业出版社 (100009 北京市西城区刘海胡同 7 号 010-66184477)
北京希望电子出版社 (100085 北京市海淀区上地 3 街 9 号金隅嘉华大厦 C 座 611)
网址: www.bhp.com.cn 电话: 010-82702660 (发行) 010-62541992 (门市)
印刷: 北京媛明印刷厂
发行: 全国新华书店经销
版次: 2006 年 12 月第 1 版
印次: 2006 年 12 月第 1 次
开本: 787mm×1092mm 1/16
印张: 15
字数: 342 千字
印数: 0001~3000 册
定价: 24.00 元

现代高职高专信息技术教材

编 委 会 成 员 名 单

主 编：高 林

副主编：鲍有文 黄春麟 丁桂芝

编 委：（按姓氏笔画排序）

丁桂芝 马小军 王 本 王 辉 付百文

申 蔚 朱立平 刘 莹 李京平 孙 悦

孙 慧 安淑芝 张俊玲 陆卫民 林志英

高 林 袁家政 曹 聪 黄春麟 鲍有文

樊月华

序 言

在新的世纪里,科学技术突飞猛进,知识经济日益显著。以微电子为基础,计算机、通信、网络、自动化、电子技术为主体的信息技术,是当前人类社会中发展最快、渗透性最强、应用面最广的先导技术。信息技术的广泛应用推动着以信息产品制造业、软件业、信息系统集成业和信息咨询服务业为主体的信息产业的发展。信息已成为重要的生产要素和战略资源,信息技术成为先进生产力的代表,信息产业将发展成为现代产业的带头产业,人类即将跨越工业时代进入信息时代。因此,信息化成为当今世界经济和社会发展的趋势,大力推进社会和国民经济信息化是推进我国社会主义现代化建设的重要任务。信息技术和产业的发展不仅需要大批专业技术人才,而且还产生了一批新的职业岗位。毋庸置疑,与信息技术相关的职业将成为未来最走俏的职业。

信息技术的人才需求将呈多元化趋势,表现在科学、工程、技术、管理、服务诸多方面的多元化。不仅需要从事信息科学、信息技术研发的人才,而且更缺少把研发成果转变为现实产品的工程化人才和产业人才,无论是从事信息科学、信息技术研发还是信息产品转化都需要大批的基础性人才和这类人才中的精英人才、领军人才。这实际上是对我国的高等教育和教育改革提出了新的要求和新的课题,要求我国的高等教育人才培养的多元化——针对社会需求,培养不同类型人才,这就要求改革我国高等教育的课程模式。在大学专科层次积极发展高等职业教育,进行技术应用型本科试点,培养工程硕士,开办软件学院都是这种尝试的重要组成部分,是高等教育适应和推动社会经济发展的重要标志。本套教材的编写就是为适应我国高等教育发展和培养技术应用型人才的需要,改革学科性的单一培养模式,在信息技术的专业领域进行教学改革的一次尝试。

本套教材的编写有以下特点:

(1) 注重技术能力目标分析,每本书都提出了该技术领域的技术能力目标,在每章开始的要点中,总结概括了其智能结构要求。

(2) 在编写过程中,注意以技术能力培养为本的课程模式的特点,先提出“问题”,然后对问题进行“分析”,最后总结归纳建立“概念”。即“找出问题,分析问题,总结归纳”是本套教材的写作特点。

(3) 案例分析是本套教材的又一写作特点,技术能力的培养不仅是要培养学生的逻辑思维能力,更要培养学生的形象思维能力,案例分析是完成这种培养的有效方法。

(4) 针对典型技术问题,有的放矢地讲解是本套教材的第四个写作特点。

(5) 技术是练出来的而不是讲出来的,给出用于训练的大量题目,使学生通过实际练习,达到掌握技术的目的,是本套教材编写的第五个写作特点。

本套教材包括《操作系统原理与应用(Windows 2000)》、《操作系统原理与应用(Unix/Linux)》、《可视化程序设计与VB》、《数据库设计技术(SQL-Server)》、《数据结构》、《网络程序设计与Java语言》、《计算机网络工程》、《计算机信息系统集成技术(Web技术)》、《多媒体实用技术》、《虚拟现实技术》等,适用于信息技术领域以技术应用性为培养目标的专业和高等职业教育专业,如计算机应用技术、通信、电子技术、自动化技

术、软件技术及其相关专业选用，也可供从事信息技术的专业人员参考或作为继续教育的培训教材。

本套教材在编写过程中，始终得到了中国高等职业教育学会、全国高等院校计算机基础教育研究会高职高专专业委员会的大力支持和帮助，并指派了优秀教师参加编写，教材编委会在此对两个学会致以诚挚的谢意。

现代高职高专信息技术教材编委会

前 言

在科学技术高速发展的信息时代，我国的计算机产业也迅猛发展，对专业人才的需求也日益迫切。因此，学习计算机知识已成为各界人士的需求，更是当代大学生知识结构中必不可少的组成部分。

“数据结构”课程是计算机程序设计的重要理论基础，是计算机学科的核心课程。“数据结构”作为一门专业基础课，要求学生能够根据计算机要处理的数据特性，为数据选择适当的逻辑结构、存储结构及其相应的算法，并初步掌握算法的时间分析和空间分析技术。

本书共由 10 章构成，较详细介绍了数据结构的概念、算法及算法分析，线性表、栈和队列的概念及应用，串、多维数组和广义表，二叉树、哈夫曼树、森林、图及其应用，排序方法以及文件等。

本书书后给出了两套综合练习题及实验内容，以方便读者学习及自测检查。

本书由林志英、付百文、朱立平编写。林志英编写了第 1、5、9、10 章；付百文编写了 6、7、8 章；朱立平编写了 2、3、4 章。鲍有文对全书进行了审校。

在成书过程中，鲍有文教授提出了很多有益的建议，给了我们很大的帮助，在此表示衷心的感谢！

由于编者水平有限，疏漏难免，敬请广大读者批评指正。

编 者

目 录

第1章 概述	1
1.1 为什么要研究数据结构	1
1.2 数据结构的概念	3
1.3 算法及算法分析	5
1.3.1 算法	5
1.3.2 算法分析	5
1.4 习题	7
第2章 线性表	8
2.1 线性表的逻辑结构	8
2.1.1 线性表的定义	8
2.1.2 线性表的基本操作	9
2.2 线性表的顺序存储结构	9
2.2.1 顺序表	9
2.2.2 在顺序表上实现基本操作	10
2.3 线性表的链式存储结构	13
2.3.1 单链表	13
2.3.2 在单链表上实现基本操作	14
2.3.3 双向链表	20
2.3.4 循环链表	24
2.4 线性表应用举例	24
2.5 习题	29
第3章 栈和队列	32
3.1 栈	32
3.1.1 栈的逻辑结构	32
3.1.2 栈的顺序存储结构	33
3.1.3 栈的链式存储结构	34
3.2 队列	36
3.2.1 队列的逻辑结构	36
3.2.2 队列的顺序存储结构	37
3.2.3 队列的链式存储结构	40
3.3 栈和队列应用举例	44
3.4 习题	51
第4章 串	53
4.1 串的逻辑结构	53
4.1.1 串的定义	53
4.1.2 串的基本操作	54

4.2 串的存储结构	54
4.2.1 串的顺序存储结构	54
4.2.2 串的链式存储结构	56
4.3 串基本操作的实现	57
4.4 习题	62
第5章 多维数组和广义表	64
5.1 多维数组	64
5.2 数组的顺序表示	64
5.3 矩阵的压缩存储	65
5.3.1 特殊矩阵	66
5.3.2 稀疏矩阵	67
5.4 广义表	72
5.4.1 广义表的定义	72
5.4.2 广义表的存储	74
5.5 习题	75
第6章 树	76
6.1 树	76
6.1.1 树的定义	76
6.1.2 树的表示方法	77
6.1.3 基本术语	78
6.1.4 树的基本操作	79
6.2 二叉树的定义、性质	79
6.2.1 二叉树的定义	79
6.2.2 二叉树的性质	80
6.3 二叉树的存储结构	82
6.3.1 二叉树的顺序存储结构	82
6.3.2 二叉树的链式存储结构	83
6.4 二叉树的遍历	84
6.4.1 二叉树的先序遍历	84
6.4.2 二叉树的中序遍历	85
6.4.3 二叉树的后序遍历	86
6.4.4 二叉树的按层次遍历	86
6.4.5 二叉树遍历的应用	89
6.5 线索二叉树	90
6.6 树和森林	91
6.6.1 树的存储结构	91

6.6.2 树、森林与二叉树的转换.....	93	8.3 效率较高的排序算法.....	144
6.6.3 树和森林的遍历.....	94	8.3.1 快速排序.....	144
6.7 哈夫曼树及其应用.....	94	8.3.2 堆排序.....	148
6.7.1 哈夫曼树的定义.....	94	8.3.3 归并排序.....	153
6.7.2 哈夫曼树的构造方法.....	95	8.3.4 排序算法的应用.....	156
6.7.3 哈夫曼编码.....	96	8.4 基数排序.....	158
6.8 综合实例——家族树的建立和统计.....	97	8.5 内部排序算法的比较和选用.....	159
6.9 习题.....	103	8.6 综合实例.....	161
第7章 图及其应用.....	106	8.7 习题.....	169
7.1 图的定义和基本概念.....	106	第9章 查找的基本概念.....	171
7.1.1 图的定义.....	106	9.1 基本概念.....	171
7.1.2 基本术语.....	107	9.2 静态查找.....	172
7.2 图的存储结构.....	108	9.2.1 顺序查找.....	172
7.2.1 邻接矩阵(二维数组)表示法.....	108	9.2.2 二分查找.....	173
7.2.2 邻接表表示法.....	110	9.2.3 分块查找.....	176
7.3 图的遍历.....	112	9.3 树表的查找.....	176
7.3.1 深度优先搜索遍历.....	112	9.3.1 二叉排序树.....	176
7.3.2 广度优先搜索遍历.....	114	9.3.2 平衡的二叉排序树.....	182
7.3.3 图的应用举例.....	115	9.4 哈希表.....	185
7.4 图的应用.....	116	9.4.1 哈希表.....	186
7.4.1 最小生成树.....	116	9.4.2 哈希函数的构造方法.....	187
7.4.2 最短路径.....	118	9.4.3 处理冲突的方法.....	188
7.4.3 有向图的拓扑排序.....	120	9.4.4 哈希表的查找及分析.....	190
7.4.4 关键路径和最短工期.....	122	9.5 习题.....	190
7.5 综合实例.....	125	第10章 文件.....	191
7.6 习题.....	130	10.1 基本概念.....	191
第8章 内部排序.....	133	10.2 顺序文件.....	192
8.1 基本概念.....	133	10.3 索引文件.....	192
8.1.1 排序.....	133	10.4 散列文件.....	194
8.1.2 排序算法的分类.....	134	10.5 多关键字文件.....	194
8.1.3 排序算法的稳定性.....	134	10.5.1 多重表文件.....	194
8.1.4 排序算法分析.....	134	10.5.2 倒排文件.....	195
8.2 简单的排序算法.....	135	10.6 习题.....	196
8.2.1 冒泡排序.....	135	附录.....	197
8.2.2 直接选择排序.....	137	综合练习题(一).....	207
8.2.3 直接插入排序.....	139	综合练习题(二).....	220
8.2.4 简单排序算法的应用.....	141		

本章知识点及重点

- 数据结构的概念
- 算法及算法分析

概 述

1.1 为什么要研究数据结构

随着计算机技术的飞速发展,计算机加工处理的对象由数值发展到字符、表格和图像等非数值数据。对于非数值数据,数据之间往往有着复杂的内在联系,单纯依靠程序设计技巧已无法编制出高效可靠的程序,而必须对数据本身的结构加以研究,不同的数据组织结构,可以直接影响到处理问题的效率和算法的设计。因此,数据结构是一门研究非数值计算的程序设计问题的学科。

著名的瑞士计算机科学家 N.Wirth 教授曾提出:算法+数据结构=程序。由此可见,程序设计的实质是对问题选择一种适合的数据结构,设计一个好的算法。

在数据处理中,为了提高处理效率,人们更关注的是数据集合中各数据元素之间存在什么关系,如何组织这些数据。下面通过一个例子来说明对同一批数据采用不同的表示方法对处理效率的影响。

【例 1-1】电话号码查询问题。

已知一本电话通讯录如表 1-1 所示,现要查找王雨的电话号码。由于通讯录中的姓名是顺序存放的,因此需要对通讯录从头到尾进行查找姓名为王雨的人,直至找到或未找到。

表 1-1 通讯录

序号	姓名	电话
1	陈 虹	85552282
2	方 波	45557788
3	方秀丽	30076545
4	黄 杨	91234651
5	黄雅玲	88601531
6	黎先生	10391231
7	李柏麟	85559482
8	林慧欣	35550297
9	林小珊	45552282
10	刘 鹏	11355555
11	刘 习	91244540
12	刘先生	35557647
13	刘小玲	30074321

续表

序号	姓名	电话
14	王 刚	35554729
15	王 瑞	30058460
16	王 雨	95554729
17	王俊元	40678888
18	王信城	15553392
19	王炫皓	55539321
20	谢丽秋	45551212
21	谢小印	76753425

如果通讯录中的数据量比较大时，此查找工作会变得非常费时和麻烦。可以采用下列方式存储通讯录：除原有的通讯录数据外，增加一个索引表，该索引表中记录每个姓氏在通讯录中的起始位置，如图 1-1 所示。采用该存储结构后，查找某人（如王雨）电话号码的工作就变为先在索引表中查找王姓在通讯录中的起始位置，结果为 14，然后在通讯录中从序号为 14 的位置开始到 19 的范围内进行查找，这样就大大缩短了查找的时间。

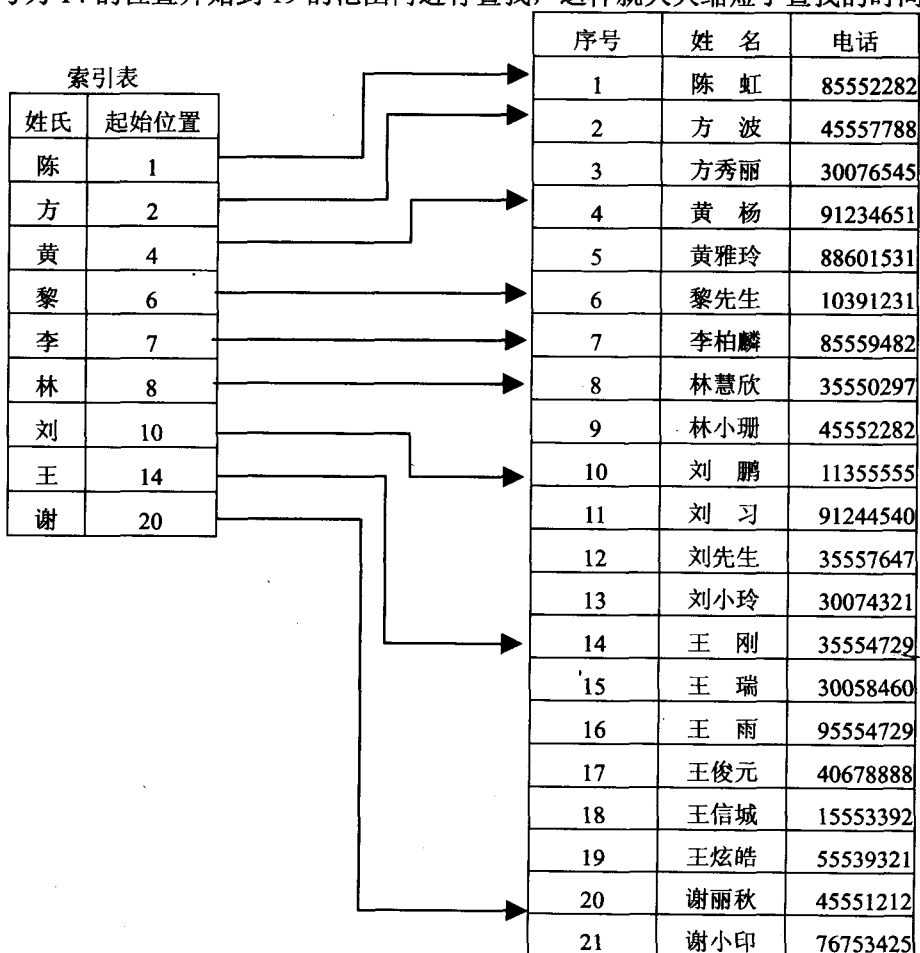


图 1-1 索引存储的通讯录

从上述例子可以看出, 选择合适的数据结构, 才能设计出有效的算法。

1.2 数据结构的概念

1. 几个基本概念

数据 (Data): 它是计算机程序加工处理的对象, 描述客观事物的数字、文字、图像、语言及其他符号的组合。数据分为两类: 数值数据, 由 0~9 数字组成; 非数值数据, 由文字、图像和其他符号组成。

数据元素: 数据元素是数据的基本单位。有时一个数据元素可以由若干个数据项组成, 在这种情况下, 常把数据元素称为记录。数据项是具有独立含义的最小单位。例如, 表 1-2 中的每一行 (即一个学生的信息) 为一个数据元素, 而其中的每一项 (如学生姓名、出生日期等) 为一个数据项。

表 1-2 学生信息表

学号	姓名	性别	出生日期	家庭住址
030335701	李丽	女	1985-04-10	北京市海淀区新新小区 2-202
030335702	王红	女	1984-12-01	北京市北四环东路 85 号
030335703	赵刚	男	1985-09-15	北京市顺义区杨镇
030335704	钱明	男	1985-12-12	北京市东城区嘉和小区 3-301
.....				

数据类型 (Data Type): 一组可能取的数据值及其定义上的操作的集合称之为数据类型。一个变量的数据类型是指该变量所有可能取值的集合及允许在这些值上的操作的集合。数据类型可分为两类:

(1) 原子数据类型。原子数据类型是不能再分的数据类型。如 C 语言中的整型、字符型等。

(2) 结构数据类型。结构数据类型可以进一步分解, 它的成分可以是原子类型, 也可以是其他结构类型。

抽象数据类型 (Abstract data type): 简称 ADT, 是指一个数学模型和在该模型上定义的操作集合。

2. 数据结构的概念

数据结构指的是数据之间的相互关系, 即数据的组织形式, 它一般包括以下 3 个方面的内容。

(1) 数据元素之间的逻辑关系, 也称为数据的逻辑结构 (Logical Structure)。

(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)。

(3) 数据的运算, 即对数据施加的操作。

数据的逻辑结构是从逻辑关系上描述数据, 它与数据的存储无关, 是独立于计算机的。数据的存储结构是逻辑结构用计算机语言的实现, 它是依赖于计算机语言的。数据的运算

是定义在数据的逻辑结构上的，每种逻辑结构都有一个运算的集合。只有确定了存储结构之后，我们才考虑如何实现这些运算。本书中讨论的数据运算，均以 C 语言描述的算法来实现。

数据的逻辑结构有两大类：线性结构和非线性结构。

(1) 线性结构。

线性结构的逻辑特征是：在数据元素的非空有限集合中，有且仅有一个开始结点；有且仅有一个终端结点；除开始结点外，集合中的每个结点只有一个前驱结点；除终端结点外，集合中的每个结点只有一个后继结点。线性表、栈、队列和字符串属于线性结构。

线性结构的基本关系图如图 1-2 所示。

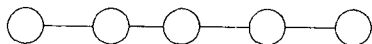


图 1-2 线性结构的基本关系图

(2) 非线性结构。

非线性结构的逻辑特征是一个结点可能有多个直接前趋和直接后继。树和图属于非线性结构。

非线性结构的基本关系图如图 1-3 和 1-4 所示。

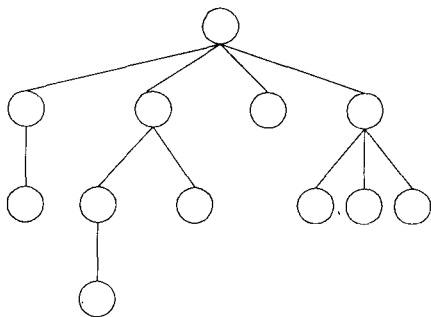


图 1-3 树形结构的基本关系图

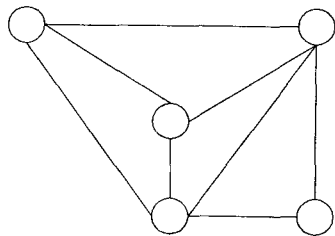


图 1-4 图状结构的基本关系图

数据的存储结构可用以下 4 种基本的存储方法得到。

(1) 顺序存储方法。

顺序存储方法是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。结点之间的逻辑关系由存储单元的邻接关系来体现。由此得到的存储结构称为顺序存储结构 (Sequential Storage Structure)。顺序存储结构通常借用程序语言的数组来描述。

(2) 链式存储结构。

链式存储方法不要求逻辑相邻的结点在物理位置上亦相邻，结点间的逻辑关系是由附加的指针来表示的。由此得到的存储结构称为链式存储结构 (Linked Storage Structure)，链式存储结构通常借助于程序语言的指针类型来实现。

(3) 索引存储方法。

索引存储方法通常是在存储结点信息的同时，还建立附加的索引表。索引表中的每一项称为索引项，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。

(4) 散列存储方法。

散列存储方法的基本思想是根据结点的关键字直接计算出该结点的存储地址。

1.3 算法及算法分析

1.3.1 算法

算法是由若干指令组成的有限序列，一个算法应具有下列 5 个重要特性。

(1) 有穷性。

一个算法必须是（对任何合法的输入值）在执行有穷步骤之后结束，且每一步都可在有穷时间内完成。

(2) 确定性。

算法中每一条指令必须有确切的含义，无二义性，即对于相同的输入只能得出相同的输出。

(3) 可行性。

算法中描述的操作都可以通过已经实现的基本运算执行有限次来实现。

(4) 输入。

一个算法有零个或多个的输入，这些输入取自于某个特定的对象的集合。

(5) 输出。

一个算法有一个或多个输出。

算法的概念和程序相似，但二者是有区别的。程序中的指令必须是机器可执行的，算法中的指令不一定是机器可以执行的。将一个算法用一个机器可以执行的语言来描述，则该算法就成为一个程序。

用于算法描述的方法有很多，有自然语言、各种高级语言、流程图以及伪码等。本书中采用 C 语言来描述算法。

1.3.2 算法分析

1. 算法评价

在解决同一个问题时，可能有多种方法，如何来评价这些算法的好坏呢？一个好的算法应达到以下目标：

(1) 正确性：算法应能正确满足具体问题的要求。

(2) 可读性：算法应易于阅读和理解，以便于调试和修改程序。

(3) 健壮性：当输入非法数据时，算法能做出适当的反映和处理，不会产生不需要的结果。

(4) 效率：算法的效率包括时间效率和空间效率。时间效率是指执行算法时所耗费的时间；空间效率是指执行算法时所耗费的存储空间，除了包括数据本身所占的存储空间外，还包括算法所需的辅助存储空间。

除了选用的算法是正确的，还要考虑算法的时间效率和空间效率。要想选择运行时间短、占用存储空间小的十全十美的算法是比较困难的，因为要节约算法的执行时间往往要

以牺牲更多的空间为代价；为了节省空间往往是以花费更多的时间为代价。因此我们只能根据具体情况有所侧重。

2. 算法分析

算法执行时间需要通过依据该算法编制的程序在计算机上运行时所耗费的时间来度量。而一个程序在计算机上运行时所耗费的时间取决于下列因素：

- (1) 问题的规模。
- (2) 依据的算法选用何种策略。
- (3) 书写程序的语言。
- (4) 机器执行指令的速度。
- (5) 对源程序编译的时间。

由于同一算法与实现算法的语言、采用的编译程序、以及所使用的计算机均有密切的联系，要精确地确定一个算法的执行时间是非常困难的。因此，在讨论算法的时间效率时，通常不考虑计算机硬件和软件等不确定因素，而用算法中所有语句的频度之和来表示一个算法所需要的时间。语句的频度 (Frequency Count) 是指该语句重复执行的次数。

例如，下面是 $n \times n$ 矩阵相乘的算法。

```
#define n 5
MatrixMlt(float a[n][n],float b[n][n],float c[n][n])
{
    int i,j,k;
    (1) for(i=0;i<n;i++)
    (2)     for(j=0;j<n;j++)
        {
    (3)         c[i][j]=0;
    (4)         for(k=0;k<n;k++)
    (5)             c[i][j]=c[i][j]+a[i][k]*b[k][j];
        }
}
```

在该算法中，每条语句执行的频度如表 1-3 所示。

表 1-3 语句执行频度

语句	语句频度
(1)	$n+1$
(2)	$n(n+1)$
(3)	n^2
(4)	$n^2(n+1)$
(5)	n^3

该算法中的时间耗费为：

$$\begin{aligned}
 T(n) &= n+1 + n(n+1) + n^2 + n^2(n+1) + n^3 \\
 &= 2n^3 + 3n^2 + 2n + 1
 \end{aligned}$$

从上式可以看出，矩阵相乘算法的执行时间是一个矩阵阶数 n 的函数。我们将问题的

输入量称为问题的规模，如矩阵相乘算法的规模是矩阵的阶数。一般情况下，算法中的执行时间是问题规模 n 的某个函数 $f(n)$ ，算法的时间度量记作：

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

它表示随问题规模 n 的增大，算法执行时间的增长率和 $f(n)$ 的增长率相同，称作算法的渐近时间复杂度 (Asymptotic Time Complexity)，简称时间复杂度。

两个矩阵相乘算法的时间量级是 $T(n) = O(n^3)$

算法常见的时间复杂度有：常数阶 $O(1)$ 、对数阶 $O(\log_2 n)$ 、线性阶 $O(n)$ 、平方阶 $O(n^2)$ 、指数阶 $O(2^n)$ 等。指数阶 $O(2^n)$ 的算法效率很低，尽量不要采用。

与算法的时间复杂度类似，一个算法的空间复杂度 (Space Complexity) 定义该算法所需的存储空间，记作：

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

它也是问题规模 n 的一个函数。

1.4 习 题

1. 什么叫数据结构？数据结构对算法有什么影响？请举例说明。
2. 什么是存储结构？什么是逻辑结构？
3. 什么是算法？算法的基本性质是什么？
4. 什么是算法的时间复杂度？什么是算法的空间复杂度，如何确定算法的时间复杂度。
5. 分析以下算法的时间性能。

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

2

线性表

本章知识点及重点

- 线性表的逻辑结构
- 线性表的顺序存储结构
- 线性表的链式存储结构：单链表、双向链表、循环链表

线性表是一种最常用、最简单的数据结构，它的结构特征好比在一条“线”上，打了若干个“结”（称为结点）。如果线上有“结”的话，表示它是一个非空表。非空线性表的特点是：只能有一个起点，而且只能有一个终点，其他的结点只能有一个直接前趋结点和一个直接后继结点。本章我们将学习线性表及其基本操作。

2.1 线性表的逻辑结构

2.1.1 线性表的定义

线性表是一个含有 n ($n \geq 0$) 个具有相同属性的数据元素的有限序列，各元素有着依次相邻的逻辑关系。

线性表中数据元素的个数 n 称为线性表的长度。当 $n=0$ 时，线性表称为空表。当 $n>0$ 时，线性表可以记为

$$(a_1, a_2, a_3, \dots, a_i, \dots, a_n)$$

线性表中的数据元素 a_i 可以是一个数、或一个符号，还可以是更加复杂的信息。例如：

(1) $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$ 是一个线性表，其中数据元素是数字，共有 10 个数据元素。

(2) $(A, B, C, \dots, Z)$ 是一个线性表，其中的数据元素是英文大写字母，共有 26 个数据元素。

(3) 表 2-1 所示的“学生成绩单”也是一个线性表，其中的数据元素是每个学生对应的一行信息（也称为“记录”），每条记录由学号、姓名、性别和成绩共 4 个数据项组成。此表中共含有 30 条记录。

表 2-1 学生成绩单

学号	姓名	性别	成绩
1001	赵	男	85
1002	钱	男	90
1003	孙	女	95
⋮	⋮	⋮	⋮
1030	李	女	85

线性表中的第一个数据元素 a_1 称为起点元素，最后一个数据元素 a_n 称为终点元素。通