



高等职业教育“十一五”规划教材

高职高专计算机专业基础系列教材

# 数据结构 (第二版)

郑泳 方风波 © 主编



免费提供电子课件



科学出版社

[www.sciencep.com](http://www.sciencep.com)

中国科学院优秀教材  
高等职业教育“十一五”规划教材

高职高专计算机专业基础系列教材

# 数据结构

(第二版)

郑泳 方风波 主编

王科 张宁 韩世芬 宋世发 副主编

科学出版社

北京

www.sciencep.com

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

010-64030303

科学出版社

北京

## 内 容 简 介

本书介绍了各种常用的数据结构及其操作,包括线性表、栈和队列、串、数组、树、图、查找和排序等。全书使用标准的C语言作为算法描述工具。

本书内容通俗易懂,侧重于应用,力求内容与应用实例相结合,并附有上机实验和实训指导,有利于提高读者分析问题和解决问题的能力。

本书可以作为高职高专院校计算机相关专业的教材,也可以作为专升本、自学考试的辅导教材。

### 图书在版编目(CIP)数据

数据结构/郑泳,方风波主编. —2版. —北京:科学出版社,2009  
(高等职业教育“十一五”规划教材·高职高专计算机专业基础系列教材)  
ISBN 978-7-03-024239-6

I. 数… II. ①郑…②方… III. 数据结构-高等学校:技术学校-教材  
IV. TP311.12

中国版本图书馆CIP数据核字(2009)第034356号

责任编辑:孙露露/责任校对:刘彦妮  
责任印制:吕春珉/封面设计:耕者设计工作室

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂印刷

科学出版社发行 各地新华书店经销

2009年4月第一版 开本:787×1092 1/16  
2009年4月第一次印刷 印张:16 1/4  
印数:1—3 000 字数:369 000

定价:25.00元

(如有印装质量问题,我社负责调换<新蕾>)

销售部电话 010-62134988 编辑部电话 010-62138978-8212

**版权所有,侵权必究**

举报电话:010-64030229; 010-64034315; 13501151303

## 前 言

“数据结构”是计算机及相关专业的一门重要的专业基础课，也是一门必修的核心课程。其课程教学要求是，学会分析研究计算机加工的数据对象的特性，以便在实际应用中选择适当的数据结构、存储结构和相应的算法，初步掌握算法的时间与空间性能的分析技巧，进行复杂程序设计的训练。

本书是为高职高专计算机及相关专业编写的教材，选材基本上覆盖了数据结构的主要内容。本书注重培养读者的实际操作能力，侧重于应用，力求内容与应用实例相结合；在介绍基本理论时，尽量做到浅显易懂，不过分追求知识的系统性和完整性；对于一些过于难懂的理论尽量省略；对各种数据结构问题多以实例来讲解，叙述上通俗易懂，可使读者加深对基本概念的理解，且有利于提高读者分析问题和解决问题的能力。

为了把高职高专学校培养技能型人才的目标落到实处，针对“数据结构”的技术性与综合性较突出的特点，我们为每章编写了相应的实验上机指导及实训项目指导。对教学内容中的每一种数据结构，其各种操作即实验，要求学生必须高质量完成；而对解决实际问题中的操作属于技能训练即实训项目，要求学生根据算法思想完成实现，达到技能训练的目的。本书中的实训项目有4个，涉及线性表、栈、串、树、图、排序、查找等数据结构的实际应用，由于课堂时间有限，读者可以根据自身的实际掌握情况，在课余选做。

本书使用标准的C语言作为算法描述工具，主要考虑的是：C语言基本上是各校教学计划中的基础语言，以C语言描述数据结构，在理解和掌握数据结构内容的同时，有利于进一步提高软件设计的能力。同时，目前各种考试，如计算机等级考试，特别是高职专升本及自学考试，均要求学生具有C语言和数据结构的基础。

全书共分9章：第1章引入数据结构与算法的一些基本概念与术语，并对算法描述及算法分析作了简要说明；第2~5章由浅入深地介绍了线性结构中的线性表、栈、队列、数组及字符串的基本定义、算法和应用；第6、7章介绍了非线性结构的树、二叉树和图；第8、9章介绍了在实际应用中使用非常广泛的查找和排序的基本算法，并进行了简单的时间、空间性能分析。每章配有相当数量的例题和习题，便于读者巩固所学知识。

本书配有电子课件等教学资源，下载网址：[www.abook.cn](http://www.abook.cn)。本书配套电子课件曾在2008年湖北省教师电教作品大赛评选活动中，荣获网络课程类二等奖；在全国第二届实践教学竞赛中，荣获多媒体课件类三等奖；在“第三届全国高等学校计算机课件评比”中荣获三等奖。

本书作为三年制的高职高专教材使用时，建议讲授 44 学时，上机实习 20 学时；作为二年制的高职高专教材使用时，可略去目录中带星号（\*）的部分，并适当精简其他内容，建议讲授 44 学时，上机实习 10 学时。

本书由郑泳、方风波任主编，王科、张宁、韩世芬、宋世发任副主编。其中，第 1 章由宋世发编写；第 2、3、6、7 章由郑泳编写；第 4 章由方风波编写；第 5 章由张宁编写；第 8 章由韩世芬编写；第 9 章及附录 A、附录 B 由王科编写。

在本书的编写过程中，得到了许多兄弟学校的大力支持与关心，特别是肖力老师的无私帮助，在此向他们表示感谢。

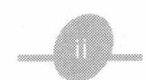
在本书的编写过程中，参考了许多相关文献，在此对文献的作者表示深深的敬意及诚挚的感谢。

由于编者水平有限，加之时间仓促，书中难免有错误和不当之处，敬请广大读者批评指正。

本书在编写过程中，参考了许多相关文献，在此对文献的作者表示深深的敬意及诚挚的感谢。由于编者水平有限，加之时间仓促，书中难免有错误和不当之处，敬请广大读者批评指正。

本书在编写过程中，参考了许多相关文献，在此对文献的作者表示深深的敬意及诚挚的感谢。由于编者水平有限，加之时间仓促，书中难免有错误和不当之处，敬请广大读者批评指正。

本书在编写过程中，参考了许多相关文献，在此对文献的作者表示深深的敬意及诚挚的感谢。由于编者水平有限，加之时间仓促，书中难免有错误和不当之处，敬请广大读者批评指正。



# 目 录

## 前言

第 1 章 概论	1
1.1 基本概念和术语	2
1.1.1 逻辑结构	2
1.1.2 存储结构	4
1.2 算法的描述与分析	5
1.2.1 算法描述	5
1.2.2 算法分析	6
1.2.3 时间复杂度	7
1.2.4 空间复杂度	9
本章小结	9
习题	9
第 2 章 线性表	12
2.1 线性表及其逻辑结构	13
2.1.1 线性表的定义	13
2.1.2 线性表的运算	13
2.2 线性表的顺序存储	14
2.2.1 顺序表结构	14
2.2.2 顺序表的基本操作	15
2.3 线性表的链式存储	20
2.3.1 单链表结构	20
2.3.2 单链表的基本操作	21
2.4 单向循环链表	26
2.5 双向循环链表	30
2.5.1 双向链表	30
2.5.2 双向循环链表	30
本章小结	31
习题	32
第 3 章 栈和队列	34
3.1 栈	35
3.1.1 栈的定义与基本运算	35

3.1.2 顺序栈 .....	35
3.1.3 链栈 .....	40
3.2 队列 .....	42
3.2.1 队列的定义及基本运算 .....	42
3.2.2 顺序队列 .....	43
3.2.3 链队列 .....	47
* 3.3 栈和队列的应用 .....	49
3.3.1 栈的应用 .....	49
3.3.2 队列的应用 .....	52
本章小结 .....	53
习题 .....	53
<b>第4章 串</b> .....	<b>55</b>
4.1 串及其运算 .....	56
4.1.1 串的基本概念 .....	56
4.1.2 串的基本运算 .....	56
4.2 串的存储结构 .....	57
4.2.1 串的顺序存储 .....	57
4.2.2 串的链式存储 .....	58
4.3 串运算的实现 .....	59
* 4.4 串的模式匹配运算 .....	62
4.4.1 有回溯的模式匹配算法 (BF 算法) .....	62
4.4.2 无回溯的模式匹配算法 (KMP 算法) .....	63
本章小结 .....	65
习题 .....	66
<b>第5章 数组和广义表</b> .....	<b>68</b>
5.1 数组 .....	69
5.1.1 数组的定义 .....	69
5.1.2 数组的顺序存储 .....	69
5.2 矩阵的压缩存储 .....	71
5.2.1 特殊矩阵 .....	71
5.2.2 稀疏矩阵 .....	74
5.3 广义表 .....	76
5.3.1 广义表的定义与运算 .....	76
5.3.2 广义表的存储 .....	77
本章小结 .....	78
习题 .....	78



第6章 树 .....	80
6.1 树的概念 .....	81
6.1.1 树的定义 .....	81
6.1.2 树的基本术语 .....	82
6.1.3 树的基本操作 .....	83
6.2 二叉树 .....	83
6.2.1 二叉树的定义 .....	84
6.2.2 二叉树的性质 .....	85
6.2.3 二叉树的存储结构 .....	88
6.3 遍历二叉树 .....	90
6.3.1 二叉树遍历方法 .....	91
6.3.2 二叉树遍历的递归算法 .....	92
* 6.3.3 二叉树遍历的非递归算法 .....	93
* 6.4 二叉树的递归编程应用 .....	95
* 6.5 线索二叉树 .....	97
6.5.1 线索二叉树的概念及存储 .....	97
6.5.2 有关线索二叉树的算法 .....	99
6.6 树和森林 .....	102
6.6.1 树的存储结构 .....	102
* 6.6.2 树、森林和二叉树的转换 .....	105
6.7 哈夫曼树及哈夫曼编码 .....	106
本章小结 .....	109
习题 .....	110
第7章 图 .....	113
7.1 图的基本概念 .....	114
7.1.1 图的定义 .....	114
7.1.2 图的基本术语 .....	115
7.2 图的存储结构 .....	117
7.2.1 邻接矩阵 .....	117
7.2.2 邻接表 .....	119
7.3 图的遍历 .....	122
7.3.1 深度优先搜索 .....	122
7.3.2 广度优先搜索 .....	125
7.4 生成树 .....	127
7.4.1 生成树 .....	127
7.4.2 最小生成树 .....	128
7.5 最短路径 .....	132



08	7.5.1 单源点最短路径	132
18	7.5.2 所有顶点对之间的最短路径	134
18	7.6 拓扑排序	135
58	7.6.1 顶点活动网	135
62	7.6.2 拓扑排序	136
68	本章小结	137
68	习题	137
	<b>第8章 排序</b>	140
68	8.1 基本概念	141
68	8.2 插入排序	142
10	8.2.1 直接插入排序	142
54	8.2.2 希尔排序	144
68	8.3 交换排序	146
68	8.3.1 冒泡排序	146
78	8.3.2 快速排序	147
78	8.4 选择排序	150
68	8.4.1 直接选择排序	150
103	8.4.2 堆排序	152
103	8.5 归并排序	154
103	8.6 排序方法的比较和选择	156
100	8.6.1 排序方法的比较	156
100	8.6.2 排序方法的选择	157
101	本章小结	157
101	习题	158
	<b>第9章 查找</b>	161
111	9.1 基本概念	162
111	9.2 线性表的查找	163
111	9.2.1 顺序查找	163
111	9.2.2 二分查找	164
110	9.2.3 分块查找	166
131	9.3 树上的查找	168
131	9.4 散列查找	174
130	9.4.1 基本概念	175
131	9.4.2 散列函数的构造方法	176
131	9.4.3 冲突处理方法	179
131	9.4.4 散列查找及分析	182
131	本章小结	182



习题 .....	183
<b>附录 A 上机实验指导</b> .....	185
实验 1 顺序表及其运算 .....	185
实验 2 链表及其运算 .....	186
实验 3 栈的运算 .....	191
实验 4 队列的运算 .....	193
实验 5 串的运算 .....	197
实验 6 二叉树的应用 .....	199
实验 7 图的存储与遍历 .....	204
实验 8 排序 .....	209
实验 9 查找 .....	213
<b>附录 B 实训项目指导</b> .....	217
实训 1 学生成绩管理 .....	217
实训 2 飞机订票系统 .....	223
实训 3 迷宫求解 .....	234
实训 4 哈夫曼编码的应用 .....	241
<b>参考文献</b> .....	249

# 概 论

## 知识教学目标

数据结构的基本概念  
数据的逻辑结构、存储结构  
算法分析

## 能力培养目标

线性表的概念、逻辑结构、存储结构  
数据的顺序存储方法和链式存储方法  
算法的定义和特性  
时间复杂度的概念和分析

随着计算机技术的快速发展,计算机的应用领域迅速扩展,它已不再局限于数值计算,而广泛地应用于数据处理、信息管理等非数值计算的各个方面。与此相应,计算机处理的数据也不再是简单的数值,而是字符串、图形、图像、语音等复杂的数据。这些复杂的数据不仅量大,而且具有一定的结构。例如,一幅图像是由一些简单的数值组成的矩阵,数据结构研究的就是这些有数据的特性、数据之间的逻辑关系、数据之间相关的运算,同时还研究这些具有逻辑关系的数据在计算机中的存储方式,并利用这些特性和关系设计出相应的算法和程序。

## 1.1 基本概念和术语

数据 (data) 是一切能被计算机识别、存储和加工处理的对象, 是对客观事务的符号表示, 是信息的载体。也就是说, 数据是对有效地输入到计算机中并能被计算机程序处理的符号的总称。随着计算机技术的发展, 数据这一观念的含义越来越广泛, 它既可以是像整数、实数、复数这样的数值数据, 又可以是像字符、文字、表格、图形、图像、声音这样的非数值数据。

数据元素 (data element) 是数据的基本单位, 相当于“记录”, 在计算机程序中要作为一个整体来考虑和处理。也就是说, 数据元素被认为是运算的基本单位, 并且具有完整、确定的实际意义。在不同条件下, 数据元素又可称为元素、顶点、记录等。例如, 职工花名册中每个人的信息就是一个数据元素。

数据对象 (data object) 是由具有相同性质的数据元素组成的集合, 是数据的一个子集, 也称为数据元素类 (data element class)。在某个具体问题中, 数据元素都具有相同的性质 (元素值不一定相等), 属于同一数据对象, 是数据对象的一个实例。例如, 整数数据对象是集合  $N = \{0, \pm 1, \pm 2, \dots\}$ 。

数据结构 (data structure) 是指数据之间相互的关系, 即数据的组织形式。它一般包括以下三个方面的内容:

1) 数据元素之间的逻辑关系, 即数据的逻辑结构。

2) 数据元素及其关系在计算机存储器中的存储方式, 即数据的存储结构, 也就是数据元素的物理结构。

3) 对数据进行的运算, 即算法。

数据的逻辑结构是从数据元素的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。因此, 数据的逻辑结构可以被看作从具体问题抽象出来的数学模型。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), 它是依赖于计算机语言的, 对机器语言而言, 存储结构是具体的。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。这些运算实际上是在抽象的数据上所施加的一系列抽象的操作。所谓抽象的操作, 是指我们只知道这些操作是“做什么”, 而无须考虑“如何做”。只有确定了存储结构之后, 我们才考虑如何具体实现这些运算。

### 1.1.1 逻辑结构

数据的逻辑结构主要有两种: 线性结构和非线性结构。为了增加感性认识, 下面举例说明。

**【例 1.1】** 职工信息管理。

假设一个职工花名册表如表 1.1 所示。花名册中每个职工的信息可由编号、姓名、性别、年龄、月收入等项目组成，占表的一行，即构成了一个数据结构。表中的每一行被看作一个结点或一个元素、一个记录，它由编号、姓名、性别、年龄、月收入等项目组成。从而整张表构成了一个关于职工花名册的数学模型，此时的花名册就是一张所谓的线性表，而计算机就可以按某个特定要求对其进行操作。表中的结点和结点之间是一种简单的线性关系。表中有且仅有一个结点为表头（第一个结点），有且仅有一个结点为表尾（最后一个结点）；对表中任意结点，与它相邻且在它前后的结点最多只有一个，这就是上述花名册的逻辑结构。

表 1.1 职工花名册

编号	姓名	性别	年龄/岁	月收入/元
1	张三	男	51	980
2	张四	女	28	880
3	王五	男	34	920
4	赵六	男	43	950
⋮	⋮	⋮	⋮	⋮

当考虑上述花名册表中的数据用计算机进行计算或处理时，就先要涉及这些结点如何在计算机中的存储表示的问题，也就是存储结构问题。对于这样的花名册表，有可能是调入一些职工时必须增加新的结点，而某些职工调离时又必须将这些相应结点从表中删除掉。究竟如何进行插入、删除、修改、查找，这就是数据的运算问题。只有弄清这些问题后，才能有效地使用花名册这个数据结构，有效地解决该单位职工基本信息的计算机辅助管理问题。

### 【例 1.2】 学校组织结构。

学校组织结构是大家熟悉的一种数据结构。它是一种非线性结构，如图 1.1 所示。我们平常所说的家族的家谱、磁盘目录结构等都属于这种结构。在这种结构中，每个结点最多有一个直接前趋，但允许有多个直接后继，结点之间存在一对多的层次关系。由于其形状像一棵倒着长的树，故称为树结构。

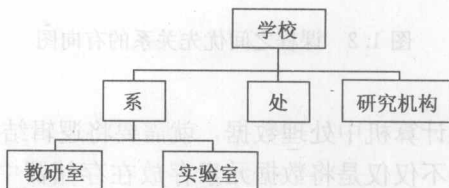


图 1.1 学校组织结构图

### 【例 1.3】 教学计划编排问题。

一个教学计划包含许多课程，如表 1.2 所示。在这些课程之间，有些又必须按规定的先后次序进行，而有些又没有次序要求，即有些课程之间有先修和后续的关系。例如，学习“计算机文化基础”，不需要先修其他课程；而在开始学习“数据结构”之前，必须先学完“计算机文化基础”和“C 语言程序设计”。这就是说，某课程的先修课程

是学习该课程的先决条件，于是，先决条件定义了各课程之间的先后关系。这种关系如图 1.2 所示。这种非线性结构称为图，其逻辑特点是：每个结点允许有多个直接前趋和多个直接后继，结构中的结点之间存在多对多的任意关系。

表 1.2 计算机专业的课程设计

课程编号	课程名称	先修课程
C <sub>1</sub>	计算机文化基础	无
C <sub>2</sub>	数据结构	C <sub>1</sub> , C <sub>4</sub>
C <sub>3</sub>	汇编语言	C <sub>1</sub>
C <sub>4</sub>	C 语言程序设计	C <sub>1</sub>
C <sub>5</sub>	计算机图形学	C <sub>2</sub> , C <sub>3</sub> , C <sub>4</sub>
C <sub>6</sub>	接口技术	C <sub>3</sub>
C <sub>7</sub>	数据库原理	C <sub>9</sub> , C <sub>2</sub>
C <sub>8</sub>	编译原理	C <sub>4</sub>
C <sub>9</sub>	操作系统	C <sub>2</sub>

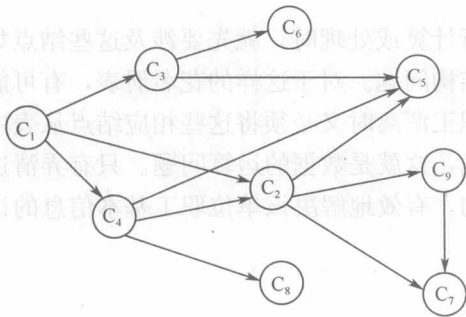


图 1.2 课程之间优先关系的有向图

### 1.1.2 存储结构

从集合论的观点出发，可以对数据结构作出形式定义：数据结构是一个二元组  $Structure = (D, R)$ 。其中，D 是数据元素的有限集合（数据对象）；而 R 是 D 上关系的有限集合。例如，图 1.2 中的逻辑结构可以表示为： $D = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9\}$ ， $R = \{ \langle C_1, C_3 \rangle, \langle C_3, C_6 \rangle, \dots, \langle C_9, C_7 \rangle \}$ 。

关系 R 描述的是数据的逻辑结构。为了在计算机中处理数据，就需要将逻辑结构存储（映像）在存储器中，即存储结构。在这里不仅仅是将数据元素存放在存储器中，而且这种存放数据元素的方法要能表现关系 R。在存储结构中，数据元素之间的关系的表示方法（即存储结构）有以下几种。

#### (1) 顺序存储

该方法采用连续的一段存储器空间存储数据元素，并借助元素在存储器中的相对位置来表示数据元素之间的逻辑关系。顺序存储方法主要应用于线性结构，非线性结构也可以通过某种线性化的方法实现顺序存储。

#### (2) 非顺序存储

该方法不要求逻辑上相邻的结点在物理位置上也相邻，结点之间的逻辑关系由附加



的“指针”来表示，即将数据元素所占的存储单元分成两部分：一部分存其值；另一部分存放表示数据元素之间关系的指针。此存储方法也称为链式存储结构。

### (3) 索引存储

该方法通常在存储结点信息的同时，建立附加的索引表。索引表中由多个索引项组成，索引项的一般形式为关键字、地址等。其中，关键字是指可唯一标识一个结点的数据项。

### (4) 哈希存储

该方法是依据结点的关键字直接计算出该结点的存储地址，而后将结点按某种方式存入该地址的一种存储方法。

一般说来，以上四种基本的存储方法，既可以单独使用，又可以组合起来使用。同一逻辑结构采用不同的存储方法，也可以得到不同的存储结构。选择何种存储结构来存储表示相应的逻辑结构，要视具体问题的要求而定。

## 1.2 算法的描述与分析

算法 (algorithm) 是程序设计的精髓，程序设计的实质就是构造解决问题的算法。它与数据结构的关系密切，在算法设计时先要确定相应的数据结构，而在讨论数据结构时也必然会涉及相应的算法。算法的设计取决于数据的逻辑结构，算法的实现取决于数据的物理结构。著名的瑞士计算机科学家 N. Wirth 所提出的公式“算法+数据结构=程序”，就深刻揭示了算法和数据结构之间的关系。对实际问题选择了一种好的数据结构之后，还得有一个好的算法，才可更好地求解问题。

### 1.2.1 算法描述

#### 1. 算法的特点

算法是对特定问题求解的一种描述。它是指令的有限序列，其中每一条指令表示一个或多个操作。此外，一个算法还具有下列 5 个重要特性。

##### (1) 有穷性

一个算法应包含有限个操作步骤，即一个算法在执行若干操作步骤之后应该能够结束，并且每一步都要在合理的时间内完成。

##### (2) 确定性

算法中的每一个步骤必须有确切的含义，无二义性，在任何情况下，对于相同的输入只能得出相同的输出。

##### (3) 可行性

算法中的每一个步骤都应该是能够通过已经实现的基本运算的有限次执行得以实现。



(4) 输入 一个算法有零个或多个输入。这些输入取自于某个选定的对象的集合。

(5) 输出

一个算法有一个或多个输出。这些输出是同输入有某些特定关系的量。  
算法的含义与程序非常相似，但又有区别。一个程序不一定满足有穷性。另一方面，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解，而程序则是算法在计算机上的特定实现。一个算法若用程序设计语言来描述，就成为一程序。

## 2. 算法的评价

算法的描述可以有多种方式，如语言方式、图形方式和表格方式等。在本书中我们采用 C 语言描述。C 语言不仅具有丰富的数据类型，也是一种较好地体现结构化程序设计原则的语言。

要设计一个好的算法，除了满足以上 5 个条件外，通常还要考虑以下几个方面。

(1) 正确性

正确性有这样 4 层含义：不含语法错误；对几组数据运行正确；对典型、苛刻的数据运行正确；对所有的数据运行正确。

(2) 可读性

可读性是指一个算法应该思路清晰、结构清楚、简单易懂。

(3) 健壮性

健壮性是指当接收到不合法输入时，算法应有适当的处理机制，以避免造成严重后果。

(4) 高效、低存储性

高效、低存储性是指在解决问题的前提下，算法应该有效地使用存储空间，且有较高的时间效率。

### 1.2.2 算法分析

要设计一个“好”的算法，首先要保证选用的算法是正确的。除此以外，还应考虑如下几个方面：

1) 执行该算法所耗费的时间。

2) 执行该算法所耗费的存储空间，其中主要考虑辅助存储空间。

3) 该算法应便于理解，易于编码、调试等。

当然，我们希望选择一个所占存储空间小、运行时间短、其他性能也好的算法。但在实际设计中，上述要求往往是相互矛盾的，要节约算法的执行时间经常要以牺牲更多的存储空间为代价；而为了节约算法所占的存储空间又可能以更多的执行时间为代价。



因此必须从实际问题出发, 根据具体情况而有所侧重。

另外, 要将一个算法转换成程序并在计算机上执行, 其运行所需要的时间还取决于下列因素:

- 1) 计算机硬件的速度。
- 2) 书写程序的高级语言。实现语言的级别越高, 其执行效率就越低。
- 3) 问题的规模。例如, 求 10 的阶乘和求 1000 的阶乘所需要的执行时间当然是不同的。

因此, 根据算法的绝对执行时间来评价算法的优劣是不合适的。为此, 可以将上述各种与计算机相关的软硬件因素都确定下来。这样, 一个特定算法的运行工作量的大小就只依赖于问题的规模 (通常用正整数  $n$  来表示), 或者说它是问题规模的函数。

评价一个算法优劣的重要依据是看这个算法执行需要占用多少机器资源。而在各种机器资源中, 时间和空间是两个最主要的方面。因此, 在进行算法评价时, 人们最关心的就是该算法在运行时所要耗费的时间代价和算法中数据结构所占用的空间代价。在这里我们分别称为时间复杂性 (所需运行时间) 和空间复杂性 (所占存储空间)。

### 1.2.3 时间复杂度

一个程序的时间复杂度 (time complexity) 是指程序从开始到结束所需要的时间。

一个算法所耗费的时间, 应该是该算法中每条语句的执行时间之和, 而每条语句的执行时间是该语句的执行次数 (也称为频度) 与该语句执行一次所需时间的乘积。但是, 当算法转换为程序之后, 每条语句执行一次所需的时间取决于机器的指令性能、速度以及编译所产生的代码质量, 这是很难确定的。我们假设每条语句所需的时间均是单位时间, 一个算法的时间耗费就是该算法中所有语句的频度之和。于是, 我们就可以独立于机器的软硬件环境来分析算法的时间耗费。

**【例 1.4】** 两个  $N \times N$  矩阵  $A$  和  $B$  的乘法, 结果为矩阵  $C$ 。

- 1) for ( $i=0$ ;  $i < N$ ;  $i++$ )  $n+1$
- 2)   for ( $j=0$ ;  $j < N$ ;  $j++$ )  $n(n+1)$
- 3)      $\{c[i][j] = 0;$   $n^2$
- 4)       for ( $k=0$ ;  $k < N$ ;  $k++$ )  $n^2(n+1)$
- 5)          $c[i][j] += a[i][k] * b[k][j];$   $n^3$

一个算法所需的运算时间通常与所解决问题的规模大小有关。通常, 我们用  $n$  作为表示问题规模的量。例如, 树的问题中  $n$  是树的顶点数; 排序问题中  $n$  为所需排序元素的个数等。我们经常把算法运行所需的时间  $T$  表示为  $n$  的函数, 记为  $T(n)$ 。

在例 1.4 中, 右边列出的是各语句的频度。该算法中所有语句的频度之和 (即算法的时间耗费) 为  $T(n) = 2n^3 + 3n^2 + 2n + 1$ 。

大部分情况下要准确地计算  $T(n)$  是很困难的, 我们往往研究所谓的“渐进时间复