



全国计算机等级考试辅导丛书

(根据最新考试大纲编写)

# 二级公共基础知识

# 实战训练教程

吕新平 李宏铭



西安交通大学出版社

XI'AN JIAOTONG UNIVERSITY PRESS



全国计算机等级考试辅导丛书  
(根据最新考试大纲编写)

# 二级公共基础知识

## 实战训练教程

吕新平 李宏铭



西安交通大学出版社  
XI'AN JIAOTONG UNIVERSITY PRESS

· 西安 ·

## 内 容 提 要

本书根据 2004 年教育部考试中心公布的《全国计算机等级考试大纲》，对于所有的二级考试公共基础知识部分的考试要求编写而成。内容包括：基本数据结构和算法、程序设计基础、软件工程基础和数据库设计基础。每一章都通过典型例题的分析对考试中的重点和难点进行了详细的分析和归纳总结，试题分析思路清晰，对参加全国计算机等级考试的考生具有实战训练之效；并在每一章的最后给出了大量的习题，这些模拟练习题针对性强，可以巩固所学的知识。书末附有 2005 年上半年和下半年全国计算机等级考试二级（公共基础知识部分）真题及参考答案。

本书可以作为全国计算机等级考试二级公共基础知识的教材和参考书，也可作为计算机相关专业的教材和参考书。

### 图书在版编目(CIP)数据

二级公共基础知识实战训练教程/吕新平,李宏铭  
编著. —西安:西安交通大学出版社,2006.2  
(全国计算机等级考试辅导丛书)  
ISBN 7-5605-2122-3

I. 二... II. ①吕... ②李... III. 电子计算机—  
水平考试—习题 IV. TP3

中国版本图书馆 CIP 数据核字(2005)第 150314 号

丛 书 名 全国计算机等级考试辅导丛书  
书 名 二级公共基础知识实战训练教程  
编 著 吕新平 李宏铭  
责任编辑 屈晓燕 贺峰涛  
出版发行 西安交通大学出版社  
地 址 西安市兴庆南路 25 号(邮编:710049)  
网 址 <http://press.xjtu.edu.cn>  
电 话 (029)82668315 82669096(总编办)  
(029)82668357 82667874(发行部)

电子信箱 [eibooks@163.com](mailto:eibooks@163.com)  
印 刷 陕西新世纪印刷厂  
字 数 193 千字  
开 本 787 mm×1 092 mm 1/16  
印 张 8.25  
版 次 2006 年 2 月第 1 版 2006 年 2 月第 1 次印刷  
印 数 0 001~3 000  
书 号 ISBN 7-5605-2122-3/TP·418  
定 价 12.00 元

---

版权所有 侵权必究

# 前 言

计算机技术是一项日新月异的高新技术,从1994年国家推出全国计算机等级考试后,计算机等级考试的内容和考核形式就在不断地变化。2004年教育部考试中心公布的《全国计算机等级考试大纲》中,就对所有二级(定位程序员)的公共基础部分作了统一的要求,内容包括基本数据结构和算法、程序设计基础、软件工程基础和数据库设计基础。考试方式为笔试,试题类型为10道选择题和5道填空题,共计30分。

本书根据2004年教育部考试中心公布的《全国计算机等级考试大纲》,对于所有的二级考试公共基础部分的考试要求编写而成。

全书共分4章。

第1章是关于数据结构与算法基础:内容包括算法的基本概念、数据结构的基本概念、线性表及其基本运算、栈和队列及其基本运算、线性链表及其基本运算、二叉树的基本概念、二叉树的存储结构、二叉树的遍历、查找技术和排序技术。

第2章是程序设计基础:内容包括程序设计方法与风格、结构化程序设计方法和面向对象的程序设计方法。

第3章是软件工程基础:内容包括软件工程基本概念、结构化分析方法、结构化设计方法、软件测试方法和程序的调试方法。

第4章是数据库设计基础:内容包括数据库系统的基本概念、数据模型、关系代数、数据库设计与管理。

本书的每一章都通过典型例题的分析对考试中的重点和难点进行了详细的分析和归纳总结,试题分析思路清晰,对参加全国计算机等级考试的考生具有实战训练之效;并在每一章的最后给出了大量的习题,这些模拟练习题针对性强,可以巩固所学的知识。

作 者

2005.12

# 目 录

<b>第 0 章 全国计算机等级考试概述</b> .....	(1)
0.1 全国计算机等级考试简介 .....	(1)
0.2 2005 年全国计算机等级考试(二级)调整说明 .....	(1)
0.3 全国计算机等级考试(二级公共基础知识)考试大纲(新版) .....	(2)
0.4 全国计算机等级考试应试指南 .....	(3)
0.5 有关参考网站简介 .....	(4)
<b>第 1 章 基本数据结构与算法</b> .....	(5)
1.1 数据结构概述 .....	(5)
1.1.1 数据结构的定义 .....	(5)
1.1.2 数据结构的图形表示 .....	(6)
1.1.3 线性结构与非线性结构 .....	(6)
1.2 算法 .....	(7)
1.2.1 算法的基本概念 .....	(7)
1.2.2 算法复杂度 .....	(9)
1.3 线性表.....	(10)
1.3.1 线性表的定义.....	(10)
1.3.2 线性表的顺序存储结构.....	(10)
1.3.3 线性表的插入运算.....	(11)
1.3.4 线性表的删除运算.....	(12)
1.4 栈和队列.....	(12)
1.4.1 栈及其运算.....	(12)
1.4.2 队列及其运算.....	(13)
1.5 线性链表.....	(15)
1.5.1 线性链表的概念.....	(15)
1.5.2 线性链表的插入运算.....	(16)
1.5.3 线性链表的删除运算.....	(17)
1.5.4 循环链表及其运算.....	(17)
1.5.5 双链表的概念.....	(18)
1.5.6 双链表的插入运算.....	(18)
1.5.7 双链表的删除运算.....	(19)
1.5.8 顺序表和链表的比较.....	(19)
1.6 树.....	(20)
1.6.1 树的基本概念.....	(20)
1.6.2 二叉树及其基本性质.....	(21)

1.6.3	二叉树的顺序存储结构	(22)
1.6.4	二叉树链式存储结构	(23)
1.6.5	二叉树的遍历	(24)
1.7	查找算法	(26)
1.7.1	查找的概念	(26)
1.7.2	顺序查找	(27)
1.7.3	二分法查找	(27)
1.8	排序算法	(27)
1.8.1	排序的概念	(27)
1.8.2	插入类排序	(28)
1.8.3	交换类排序	(29)
1.8.4	选择类排序	(30)
1.8.5	各种排序方法的比较	(31)
1.9	典型试题分析	(31)
	一、选择题	(31)
	二、填空题	(36)
1.10	模拟试题	(39)
	一、选择题	(39)
	二、填空题	(45)
1.11	模拟试题参考答案	(46)
	一、选择题	(46)
	二、填空题	(46)
<b>第2章</b>	<b>程序设计基础</b>	<b>(48)</b>
2.1	程序设计方法与风格	(48)
2.1.1	程序设计方法学	(48)
2.1.2	程序设计风格	(48)
2.2	结构化程序设计	(49)
2.2.1	结构化程序设计的原则	(49)
2.2.2	结构化程序的基本结构与特点	(50)
2.2.3	结构化程序设计原则和方法的应用	(51)
2.3	面向对象的程序设计	(51)
2.3.1	关于面向对象的方法	(51)
2.3.2	面向对象方法中的常用基本概念	(52)
2.4	典型试题分析	(53)
	一、选择题	(53)
	二、填空题	(54)
2.5	模拟试题	(55)
	一、选择题	(55)

二、填空题·····	(56)
2.6 模拟试题参考答案·····	(57)
一、选择题·····	(57)
二、填空题·····	(57)
<b>第3章 软件工程基础</b> ·····	(59)
3.1 软件工程基本概念·····	(59)
3.1.1 软件定义与软件特点·····	(59)
3.1.2 软件危机与软件工程·····	(60)
3.1.3 软件工程过程与软件生命周期·····	(61)
3.1.4 软件开发工具与软件开发环境·····	(64)
3.2 软件需求分析·····	(65)
3.2.1 结构化分析方法·····	(65)
3.2.2 结构化分析的常用工具·····	(65)
3.2.3 软件需求规格说明书·····	(67)
3.3 软件设计·····	(67)
3.3.1 软件设计的基本概念·····	(67)
3.3.2 概要设计·····	(68)
3.3.3 详细设计·····	(70)
3.4 软件测试·····	(72)
3.4.1 软件测试的目的·····	(72)
3.4.2 软件测试的原则·····	(72)
3.4.3 软件测试技术与方法·····	(72)
3.5 程序的调试·····	(76)
3.5.1 程序调试的基本概念·····	(76)
3.5.2 软件调试方法·····	(76)
3.6 典型试题分析·····	(77)
一、选择题·····	(77)
二、填空题·····	(81)
3.7 模拟试题·····	(84)
一、选择题·····	(84)
二、填空题·····	(89)
3.8 模拟试题参考答案·····	(91)
一、选择题·····	(91)
二、填空题·····	(91)
<b>第4章 数据库设计基础</b> ·····	(94)
4.1 数据的基本概念·····	(94)
4.1.1 数据、数据库、数据库管理系统·····	(94)

4.1.2	数据库系统的发展	(96)
4.1.3	数据库系统的基本特点	(96)
4.2	数据模型	(97)
4.2.1	数据模型的基本概念	(97)
4.2.2	E-R 模型	(98)
4.2.3	基本数据模型	(99)
4.3	关系代数	(100)
4.3.1	关系模型的基本操作	(100)
4.3.2	关系模型的基本运算	(101)
4.3.3	关系数据库的数据体系结构	(101)
4.3.4	关系模型的完整性规则	(102)
4.3.5	关系数据库规范化理论	(103)
4.4	数据库设计与管理	(104)
4.4.1	数据库设计的内容	(104)
4.4.2	数据库设计的需求分析	(104)
4.4.3	数据库概念设计	(105)
4.4.4	数据库的逻辑设计	(105)
4.4.5	数据库的物理设计	(106)
4.4.6	数据库管理	(106)
4.5	典型试题分析	(107)
	一、选择题	(107)
	二、填空题	(110)
4.6	模拟试题	(113)
	一、选择题	(113)
	二、填空题	(118)
4.7	模拟试题参考答案	(119)
	一、选择题	(119)
	二、填空题	(120)

<b>附录 1</b>	<b>2005 年上半年全国计算机等级考试二级(公共基础知识部分)笔试试题及参考答案</b>	(121)
-------------	--	-------

<b>附录 2</b>	<b>2005 年下半年全国计算机等级考试二级(公共基础知识部分)笔试试题及参考答案</b>	(123)
-------------	--	-------



# 第 0 章 全国计算机等级考试概述

## 0.1 全国计算机等级考试简介

全国计算机等级考试(national computer rank examination,简称 NCRE),是经教育部批准,由教育部考试中心于 1994 年开始面向社会推出的考试系统,用于考查应试人员计算机应用知识与能力的等级水平考试。

考试分为笔试和上机两部分。

NCRE 目前(2005 年之前)共设置了四个等级:

- 一级:为满足人们在一般性工作中对计算机的应用而设置的,重点是操作能力的考核。考核内容主要包括微型计算机基础知识和使用办公自动化软件及因特网(Internet)的基本技能。

- 二级:为计算机程序员而设置的。考核内容主要包括计算机基础知识和使用一种高级计算机语言(包括 QBASIC、C、FORTRAN、FoxBASE、Visual Basic、Visual FoxPro 等)编写程序以及上机调试的基本技能。

- 三级:为“开发工程师”而设置的,重点在设计、管理、测试和技术支持。三级分为“PC 技术”、“信息管理技术”、“数据库技术”和“网络技术”等四个类别。“PC 技术”考核 PC 机硬件组成和 Windows 操作系统的基础知识以及 PC 机使用、管理、维护和应用开发的基本技能。“信息管理技术”考核计算机信息管理应用基础知识及管理信息系统项目和办公自动化系统项目开发、维护的基本技能。“数据库技术”考核数据库系统基础知识及数据库应用系统项目开发和维护的基本技能。“网络技术”考核计算机网络基础知识及计算机网络应用系统开发和管理的的基本技能。“PC 技术”上机考查汇编语言编程,其他几个类别上机考查 C 语言编程。三级证书获得者应具备其对应类别的硬件或软件系统的开发和维护的基本能力。

- 四级:为“系统设计工程师和项目主管”而设置。考核内容包括计算机原理、数据结构、离散数学、操作系统、软件工程、数据库、计算机体系结构、计算机网络与通讯等。上机考试考查 C 语言编程。

NCRE 每年举行考两次,每年上半年 4 月的第一个星期六,考一、二、三级;每年下半年 9 月的倒数第二个星期六,开考一、二、三、四级。

## 0.2 2005 年全国计算机等级考试(二级)调整说明

从 2005 年开始,对计算机等级考试进行调整,关于二级部分的调整如下:

- 新增科目:新增二级 Java、二级 Access、二级 C++三个科目。

- 停考科目:二级 FORTRAN、二级 QBasic、二级 FoxBASE。

- 科目名称:对二级科目名称进行规范,根据应用性质和科目特点,将现有科目分成二级语言程序设计(C、C++、Java、Visual Basic、QBasic、FORTRAN)和二级数据库程序设计(FoxBASE、Visual FoxPro、Access)两类。

- 考核内容:二级仍然为程序员而设置,考核内容主要包括基础知识和程序设计。所有科目对基础知识做统一要求,使用统一的基础知识大纲和教程。二级基础知识主要涉及数据结构与算法、程序设计方法、软件工程、数据库基础知识共四个部分。二级基础知识在各科笔试中的比重为 30%(30 分),题型为 10 个选择题和 5 个填空题。二级上机考试中将取消对 DOS 部分的考核(占 30 分)。

- 考试形式:二级所有科目的考试形式仍为笔试和上机考试两部分。

## 0.3 全国计算机等级考试(二级公共基础知识) 考试大纲(新版)

### 基本要求

1. 掌握算法的基本概念。
2. 掌握基本数据结构及其操作。
3. 掌握基本排序和查找算法。
4. 掌握逐步求精的结构化程序设计方法。
5. 掌握软件工程的基本方法,具有初步应用相关技术进行软件开发的能力。
6. 掌握数据库的基本知识,了解关系数据库的设计。

### 考试内容

#### 一、基本数据结构与算法

1. 算法的基本概念;算法复杂度的概念和意义(时间复杂度与空间复杂度)。
2. 数据结构的定义;数据的逻辑结构与存储结构;数据结构的图形表示;线性结构与非线性结构的概念。
3. 线性表的定义;线性表的顺序存储结构及其插入与删除运算。
4. 栈和队列的定义;栈和队列的顺序存储结构及其基本运算。
5. 线性单链表、双向链表与循环链表的结构及其基本运算。
6. 树的基本概念;二叉树的定义及其存储结构;二叉树的前序、中序和后序遍历。
7. 顺序查找与二分法查找算法;基本排序算法(交换类排序,选择类排序,插入类排序)。

#### 二、程序设计基础

1. 程序设计方法与风格。
2. 结构化程序设计。
3. 面向对象的程序设计方法,对象,方法,属性及继承与多态性。

#### 三、软件工程基础

1. 软件工程基本概念,软件生命周期概念,软件工具与软件开发环境。

2. 结构化分析方法,数据流图,数据字典,软件需求规格说明书。
3. 结构化设计方法,总体设计与详细设计。
4. 软件测试的方法,白盒测试与黑盒测试,测试用例设计,软件测试的实施,单元测试、集成测试和系统测试。
5. 程序的调试,静态调试与动态调试。

#### 四、数据库设计基础

1. 数据库的基本概念:数据库,数据库管理系统,数据库系统。
2. 数据模型,实体联系模型及 E-R 图,从 E-R 图导出关系数据模型。
3. 关系代数运算,包括集合运算及选择、投影、连接运算,数据库规范化理论。
4. 数据库设计方法和步骤:需求分析、概念设计、逻辑设计和物理设计的相关策略。

#### 考试方式

1. 公共基础的考试方式为笔试,与 C 语言程序设计(C++ 语言程序设计、Java 语言程序设计、Visual Basic 语言程序设计、Visual FoxPro 数据库程序设计、Access 数据库程序设计)的笔试部分合为一张试卷。公共基础部分占全卷的 30 分。
2. 公共基础知识有 10 道选择题和 5 道填空题。

## 0.4 全国计算机等级考试应试指南

### 1. 认真学习考试大纲

准备参加计算机等级考试,应认真学习考试大纲,明确考试大纲中对考试的范围和掌握程度的要求,这样可以在复习准备的过程中抓住重点,有的放矢,并且不会遗漏应该了解的知识点。

在准备考试时,依据考试大纲所要求的知识点,根据教科书认真准备每一部分的内容。选做大量的习题,通过习题理解书中的内容,考生应重视全真题,如去年 4 月或 9 月的考试题及教育部考试中心新近出版的相关资料,这些都将是有益于在复习中把握知识点,了解到底是要考哪些内容。并且每年计算机等级考试的题目都是从题库里抽出来的,所以每次考试大概会有 10% 的重复试题出现,因此全真题的重要性就不言而喻了。

### 2. 选择适当的复习资料

准备考试时,要根据自己的实际情况,选用适当的应试资料。资料分教科书、指导书、题集三类。教科书是系统地讲授该门课程,指导书是提纲挈领地讲述该门课,题集则是知识点的一些具体形式。在准备二级考试时,若已经系统学习了 C 语言,就可选用指导书,这样便于较快地复习知识体系,掌握知识重点,提高复习的效率。若要从头学习 C 语言,则要用教科书。

### 3. 复习时做好上机练习

对于二级考试来说,计算机程序设计是一种实践性很强的课程,因此在复习的过程中,必须多注意上机训练,在实践中理解和体会程序设计的内涵并将其融会贯通。上机编程才是掌握编程语言的最佳途径。另外上机题中考的主要还是编程题,对上机成绩有举足轻重的作用。如果只是纸上谈兵,没有实际的编程经验,是很难通过的。上机题一般都有输出结果,而要产生输出结果,必须运行程序。运用程序调试方法,在运行中调试程序,是一项重要的技能。

#### 4. 考试时的注意事项

笔试的选择题是四选一,不选不得分,但选错不扣分。因此考试时不应该漏选。对于有把握的题目要保证一次答对,不要想再次验证;对于似是而非的题目,可根据经验选择一个初步答案,做一个标记,试卷完成后,有时间再重做这些带有标记的题目。考试中切记不要对个别题目花费过多的时间,影响其他题目。

笔试时的填空题要仔细考虑后再做解答,因为有些题目可能有多个答案,只要填对一个就可以了。对于填空题,不论是有把握的,还是有可能有把握的,都要认真填写。

## 0.5 有关参考网站简介

### 1. 全国计算机等级考试网

网址:<http://www.ncrc.cn>

“全国计算机等级考试网”是由教育部考试中心和北京华夏大地教育网联合推出的,其宗旨是利用互联网技术,通过网络形式为考生学习和备考提供高效、规范、适时的助学和指导服务,并发挥网络优势,进一步宣传、推广全国计算机等级考试,普及计算机教育。

### 2. 考网

网址:<http://www.kaowang.com>

考网主要提供与全国计算机等级考试、全国计算机软件水平考试相关的信息及服务,包括知识点学习、模拟考试、在线答疑等优秀栏目,其中大部分免费提供,但为了保障服务质量,自2003年7月份起,考网开始提供小额付费服务,目标是帮助考生高效复习备考。

### 3. 东方教育网

网址:<http://www.eastedu.com.cn>

东方教育网由全国计算机等级考试、全国计算机应用技术考试无锡伊斯特考点投资建设的一个以全国计算机等级考试、全国计算机应用技术考试、全国英语等级考试等各类证书考试为核心的教育考试服务类网站,是国内最优秀的、在广大考生当中具有广泛影响力和知名度的专业考试站点之一,现在的核心栏目包括政策中心、试题中心、培训中心、服务中心,其中的试题中心、主考谈考试、考试技巧集锦等访问量超前。

### 4. 无忧考网

网址:<http://www.51test.net>

中国最大在线招考资源搜索咨询网站——“无忧考网 51Test.NET”是向会员提供网络信息在线咨询服务的著名专业网站。无忧考网依托北京著名高校信息资源,信息发布权威迅捷。

### 5. 考试资源

网址:<http://www.examres.com>

考试资源网是一家以提供最新考试信息资料,方便广大参加考试的朋友而开设的个人网站,受到了广大网友的欢迎和好评。考试资源网与华夏大地教育网等著名网上学校合作,推出了一系列网上辅导课程。考试资源网站目前设立了如下栏目:信息中心、资料中心、试题中心、考试论坛、在线书店等栏目。

# 第1章 基本数据结构与算法

## 1.1 数据结构概述

随着计算机技术的发展,其应用范围在不断扩大。计算机所处理的数据量在不断扩大,所处理的数据类型已不再是单纯的数值数据,而更多的是非数值数据,如表格、声音、图形和图像等具有一定结构的数据。也就是说,这些非数值数据并不是杂乱无章的,它们一定有内在的联系,只有弄清楚它们之间的本质的联系,才能使用计算机对大量的数据进行有效的处理。数据结构就是要描述数据之间的内在联系:数据的逻辑结构、存储结构及对数据的基本操作,算法设计和算法分析等。对于程序设计来说,无论开发系统软件还是应用软件,其核心是数据结构及其算法。所以,作为软件设计技术的理论基础,“数据结构”就不仅仅是计算机学科的核心课程,也应该是所有应用计算机的其他学科所必须掌握的课程。

### 1.1.1 数据结构的定义

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,数据的范畴包括:数值、字符串、图像和声音等各种多媒体信息。

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、记录。如一年四季的季节名可以作为季节的数据元素,表示家庭成员的各成员名(父亲、母亲、儿子、女儿)可以作为家庭成员的数据元素。

数据项:一个数据元素可以由若干个数据项(也可称为字段、域、属性)组成。数据项是具有独立含义的最小标识单位。

数据结构(data structure)指的是数据之间的相互关系,即数据的组织形式。数据结构一般包括以下三方面内容:

#### 1. 数据的逻辑结构(logical structure)

数据的逻辑结构是指数据元素之间的逻辑关系。

数据的逻辑结构是从逻辑关系上描述数据,与数据的存储无关,是独立于计算机的。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. 数据的存储结构(storage structure)

数据的存储结构是指数据元素及其关系在计算机存储器内的表示。

数据的存储结构是逻辑结构用计算机语言的实现(亦称为映象),它依赖于计算机语言。对机器语言而言,存储结构是具体的。一般只在高级语言的层次上讨论存储结构。

### 3. 数据的运算

数据的运算即对数据施加的操作。

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。最常用的检索、插入、删除、更新、排序等运算实际上只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,才考虑如何具体实现这些运算。

#### 1.1.2 数据结构的图形表示

一个数据结构可以用二元关系来表示 $(D,R)$ ,其中 $D$ 表示数据集中的每一个数据元素, $R$ 表示这些数据元素之间的关系。另外一个数据结构也可直观地用图形来表示,在数据结构的图形表示中,对于数据集合 $D$ 中的每一个元素用中间标有元素值的方框(或圆框)来表示,一般称为数据结点(或简称结点),对于关系 $R$ 中的每一个二元组,用一条有向线段从直接前驱指向直接后继。例如,一年四季的数据结构可以用图1-1来表示。



图 1-1 一年四季数据结构的图形表示

#### 1.1.3 线性结构与非线性结构

在不产生混淆的前提下,常将数据的逻辑结构简称为数据结构。数据的逻辑结构有两大类:

##### 1. 线性结构

线性结构的逻辑特征是:若结构是非空集,则有且仅有一个开始结点和一个终端结点,并且所有结点都最多只有一个直接前趋和一个直接后继。也就是数据元素之间存在一对一的关系。

线性表是一个典型的线性结构。栈、队列、串等都是线性结构。

##### 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。数组、广义表、树和图等数据结构都是非线性结构。

数据的存储结构可用以下四种基本存储方法得到:

##### 1. 顺序存储方法

该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

该方法不要求逻辑上相邻的结点在物理位置上亦相邻,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

该方法通常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)

关键字是能唯一标识一个结点的那些数据项。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

该方法的基本思想是:根据结点的关键字直接计算出该结点的存储地址。

## 1.2 算法

数据的运算通过算法(algorithm)来描述,因此讨论算法是数据结构课程的重要内容之一。

### 1.2.1 算法的基本概念

#### 1. 算法的定义

当我们要编写一个程序的时候,我们总要首先想好这个程序是干什么的?应该如何实现这些目标?应该先进行什么处理,后进行什么处理?所处理的数据的格式是什么?遇到一些复杂的问题,我们可能还需要考虑采用什么数学方法来处理。这一切都涉及一个专业名词——“算法”。

算法是任意一个定义的计算过程,它以一个或多个值作为输入,并产生一个或多个值作为输出。因此我们可以从下面两个方面来理解算法:

- (1) 一个算法可以被认为是在用来解决一个计算问题的工具。
- (2) 一个算法是一系列将输入转换为输出的计算步骤。

若一个算法对于每个输入实例均能终止并给出正确的结果,则称该算法是正确的。正确的算法解决了给定的计算问题。

一个不正确的算法是指对某些输入实例不终止,或者虽然终止但给出的结果不是所期望得到的答案,一般只考虑正确的算法。

很多时候,程序设计者所面临的问题就是寻找一个合适的算法。例如,一个熟练的程序员要设计一个下“五子棋”的游戏程序,对他而言,C语言的编程规则已经清楚。他所面对的核心问题是寻找一种可以模拟人下棋的算法。因此,算法在软件设计中具有重要的地位。正如著名的计算机科学家沃思(Nikiklaus Wirth)所指出的如下公式:

程序 = 数据结构 + 算法

## 2. 算法的特性

每一个算法具有下列五个特性:

### (1) 有穷性(finiteness)

一个算法要在有限的步骤内解决问题(这里所说的步骤是指计算机执行步骤)。计算机程序不能无限地运行下去(甚至不能长时间地运行下去),所以一个无限执行的方法不能成为程序设计中的“算法”。

例如,求某一自然数  $N$  的阶乘:

$$N! = 1 \times 2 \times 3 \times \dots \times N$$

这是一个算法。因为对任何一个自然数而言,无论这个数多大,总是有限的。用这个公式计算  $N!$  总是需要有限的步骤。

但是,以下计算公式则不能作为算法,因为其计算步骤是无限的:

$$\text{SUM} = 1 + 1/1 + 1/2 + 1/3 + \dots + 1/n + \dots$$

事实上有穷性是指合理的范围之内,比如设计了一个算法是有限的,但按照目前计算机发展的水平要计算1000年才能完成,这样的算法没有实际意义,可以不当作算法,可以视为无穷。

实际上,在计算机的许多加密算法中,可以解密的方法不是不存在,而是要执行这样的解密算法需要大量的时间。这样就实现了保密。所谓保密就是让在一定的时间内信息不被他人知晓。

当然,计算机技术的进步会对算法产生影响。对于现在的计算机1000年才能完成的算法将来可能在几个月内就能完成,到那时某些现在无穷性的算法将变成切实可行的算法。

### (2) 确定性(definiteness)

算法中的每一条指令必须有确切的含义,不会产生二义性。

### (3) 可行性(effectiveness)

一个算法是能执行的,也就是说,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。

### (4) 输入

输入就是从外界取得必要的信息。一个算法可以有零个或多个输入,例如:输入一个年份,判断其是否是闰年。同时一个算法可以没有输入,例如:计算出  $5!$  是多少。

### (5) 输出

算法的目的就是求解,“解”就是我们想要得到的最终结果。输出是同输入有着某些特定关系的量。一个算法得到的最终结果就是输出。没有输出的算法是没有意义的。

## 3. 算法的描述

一个算法可以用自然语言、计算机程序语言或其他语言来说明,唯一的要求是该说明必须精确地描述计算过程。一般而言,描述算法最合适的语言是介于自然语言和程序语言之间的伪语言。它的控制结构往往类似于 Pascal、C、Java 等程序语言,但其中可使用任何表达能力强的方法使算法表达更加清晰和简洁,而不至于陷入具体的程序语言的某些细节。

## 4. 算法的设计要求

在进行算法的设计时,应满足下列的要求:



### (1) 正确性

一个算法应当能够解决具体问题。其“正确性”可分为以下几个方面：

- 不含逻辑错误；
- 对于几组输入数据能够得出满足要求的结果；
- 对于精心选择的典型、苛刻的输入数据都能得到要求的结果；
- 对于一切合法的输入都能输出满足要求的结果。

### (2) 可读性

算法应该可以用能够被人理解的形式表示。太复杂的、不能被程序员所理解的算法难以在程序设计中采用。

### (3) 健壮性

健壮性指算法具有抵御“恶劣”输入信息的能力。当输入数据非法时，算法也能适当地做出反应或进行处理，而不会产生莫名其妙的输出结果。例如，当输入三个边的长度值来计算三角形的面积时，一个有效的算法是在三个输入数据不能构成一个三角形时，应报告输入的错误，应返回一个表示错误或错误性质的值并中止执行。

### (4) 效率与低存储量的需求

高效率 and 低存储量是优秀程序员追求的目标。效率指的是算法执行时间，对于一个问题如果有多个算法可以解决，执行时间短的算法效率高。存储量需求指算法执行进程所需要的最大存储空间。效率与低存储量需求这两者都与问题规模有关。占用存储量最小、运算时间最少的算法就是最好的算法。但是在实际中，运行时间和存储空间往往是一对矛盾，要根据具体情况选择更优先考虑哪一个因素。

## 1.2.2 算法复杂度

### 1. 算法的时间复杂度

算法求解问题的输入量称为问题的规模(Size)，一般用一个整数表示。一个算法的时间复杂度(time complexity, 也称时间复杂性) $T(n)$ 是该算法的时间耗费, 是该算法所求解问题规模  $n$  的函数。当问题的规模  $n$  趋向无穷大时, 时间复杂度  $T(n)$  的数量级(阶)称为算法的渐进时间复杂度, 或简称为时间复杂度。

最坏情况下的时间复杂度称为最坏时间复杂度。一般不特别说明, 讨论的时间复杂度均是最坏情况下的时间复杂度。平均时间复杂度是指所有可能的输入实例均以等概率出现的情况下, 算法的期望运行时间。

常见的复杂度按数量级递增排列依次为: 常数  $O(1)$ 、对数阶  $O(\log_2 n)$ 、线性阶  $O(n)$ 、线性对数阶  $O(n \log_2 n)$ 、平方阶  $O(n^2)$ 、立方阶  $O(n^3)$ 、...、 $k$  次方阶  $O(n^k)$ 、指数阶  $O(2^n)$ 。显然, 时间复杂度为指数阶  $O(2^n)$  的算法效率极低, 当  $n$  值稍大时就无法应用。

### 2. 算法空间复杂度

类似于时间复杂度, 一个算法的空间复杂度(space complexity) $S(n)$ 定义为该算法所耗费的存储空间, 它也是问题规模  $n$  的函数。渐近空间复杂度也常常简称为空间复杂度。算法的时间复杂度和空间复杂度合称为算法的复杂度。