

新大纲

NORIE



National Computer Rank Examination

全国计算机等级考试

教程 二级 公共基础知识

全国计算机等级考试教材编写组 编著
未来教育教学与研究中心



- 一学就会的教程：将复杂问题简单化，将理论知识通俗化
- 衔接考试的教程：分析命题规律，考核的要点就是我们讲解的重点
- 书盘结合的教程：提供多媒体教学光盘，丰富您的学习方式
- 视频课堂：动画演绎，视频讲解，把等考辅导老师请回家，让学习变得更高效、更轻松
- 模拟软件：模拟真实考试环境，题量超大，智能评分，深受广大考生欢迎

人民邮电出版社
POSTS & TELECOM PRESS



新大纲

NORCE

Future
未来教育

National Computer Rank Examination

全国计算机等级考试

教程 二级
公共基础知识

全国计算机等级考试教材编写组
未来教育教学与研究中心 编著

人民邮电出版社
北京

图书在版编目 (CIP) 数据

全国计算机等级考试教程. 二级公共基础知识 / 全国计算机等级考试教材编写组, 未来教育教学与研究中心编著. —北京: 人民邮电出版社, 2009. 1
ISBN 978-7-115-19057-4

I. 全… II. ①全…②未… III. 电子计算机—水平考试—教材 IV. TP3

中国版本图书馆CIP数据核字 (2008) 第174654号

内 容 提 要

本书依据教育部考试中心最新发布的《全国计算机等级考试大纲》以及作者多年对等级考试的研究编写而成, 旨在帮助考生 (尤其是非计算机专业的初学者) 学习相关内容, 顺利通过考试。

全书共 4 章, 主要内容包括: 数据结构与算法 (算法与数据结构的基本概念、线性表、栈、队列、树、查找技术、排序技术)、程序设计基础 (程序设计方法与风格、结构化程序设计、面向对象程序设计)、软件工程基础 (软件工程的基本概念、结构化分析方法、结构化设计方法、软件测试及程序调试)、数据库设计基础 (数据库的基本概念、数据模型、关系代数、数据库的设计与管理)。

本书配套光盘中提供了多媒体课堂, 以动画的方式讲解重点和难点, 为考生营造一种轻松的学习环境。

本书可作为全国计算机等级考试二级培训教材和自学用书。

全国计算机等级考试教程——二级公共基础知识

- ◆ 编 著 全国计算机等级考试教材编写组
未来教育教学与研究中心
责任编辑 李 莎
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
- ◆ 开本: 880×1092 1/16 彩插: 1
印张: 7.5 2009 年 1 月第 1 版
字数: 193 千字 2009 年 1 月河北第 1 次印刷

ISBN 978-7-115-19057-4/TP

定价: 19.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

丛书序

全国计算机等级考试由教育部考试中心主办,是国内影响最大、参加考试人数最多的计算机水平考试。它的根本目的在于以考促学,这决定了它的报考门槛较低,考生不受年龄、职业、学历等背景的限制,任何人均可根据自己学习和使用计算机的实际情况,选考不同级别的考试。

一、为什么编写本丛书

计算机等级考试的准备时间短,一般从报名到参加考试只有近4个月的时间,留给考生的复习时间有限,并且大多数考生是非计算机专业的学生或社会人员,基础比较薄弱,学习起来比较吃力。

通过对考试的研究和对数百名考生的调查分析,我们逐渐摸索出一些减少考生(尤其是初学者)学习困难的方法,以帮助考生提高学习效率和学习效果。因此我们编写了本套图书,将我们多年研究出的教学和学习方法贯穿全书,帮助考生巩固所学知识,顺利通过考试。

二、丛书特色

1. 一学就会的教程

本套图书的知识体系都经过巧妙设计,力求将复杂问题简单化,将理论难点通俗化,让读者一看就懂,一学就会。

- 针对初学者和考生的学习特点和认知规律,精选内容,分散难点,降低台阶。
- 例题丰富,深入浅出地讲解和分析复杂的概念和理论,力求做到概念清晰、通俗易懂。
- 采用大量插图,并通过生活化的实例,将复杂的理论讲解得生动、易懂。
- 精心为考生设计学习方案,设置各种栏目引导和帮助考生学习。

2. 衔接考试的教程

我们深入分析和研究历年考试真题,结合考试的命题规律选择内容,安排章节,坚持多考多讲、少考少讲、不考不讲的原则。在讲解各章节的内容之前,都详细介绍了考试的重点和难点,从而帮助考生安排学习计划,做到有的放矢。

3. 书盘结合的教程

本丛书所配的光盘主要提供两部分内容:多媒体课堂、笔试与上机考试模拟系统。使用了本丛书的光盘,就等于把辅导老师请回了家。

多媒体课堂用动画演绎复杂的理论知识,用视频讲解各种操作方法,使学习变得轻松而高效。

在笔试与上机考试模拟系统中提供大量的练习题,其中上机考试模拟系统可真实模拟上机考试环境,帮助考生提前感受上机考试的全过程。

三、如何学习本丛书

1. 如何学习每一章

每章都安排了章前导读、本章评估、学习点拨、本章学习流程图、知识点详解、复习题、学习效果自评表等固定板块。下面就详细介绍如何合理地利用这些资源。

章前导读

列出每章知识点,让考生明确学习内容,做到心中有数。

章前导读

通过本章,你可以学习到:

- 什么是算法?它包含哪些复杂度
- 什么是数据的逻辑结构和存储结构
- 栈和队列的定义是什么
- 二叉树的定义是什么?有哪些性质?二叉树是如何遍历的

学习点拨

提示每章内容的重点和难点,为考生介绍学习方法,使考生更有针对性地学习。

学习点拨

本章主要介绍算法、数据结构的基础知识。读者在学习的过程中要通过对相关概念的对比理解它们之间的区别和联系。

本章评估

通过分析数套历年笔试和上机考试的真题,总结出每章内容在考试中的重要程度、考核类型、所占分值,以及建议学习时间等重要参数,使考生可以更加合理地制订学习计划。

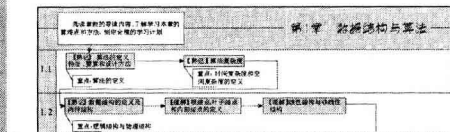
本章评估

重要度	★★★★
知识类型	理论
考核类型	笔试

本章学习流程图

提炼重要知识点,详细点明各知识点之间的关系,同时指出对每一个知识点应掌握的程度:是了解,是熟记,还是掌握。

本章学习流程图



知识点详解

根据考试的需要,合理取舍,精选内容,结合巧妙设计的知识板块,使考生迅速把握重点,顺利通过考试。

1.1 算法

本节从算法的基本概念入手,阐述算法的基本特征、基本要素、设计方法以及设计准则,进而详细讲解算法的时间复杂度和空间复杂度。

1.1.1 什么是算法

算法的定义

有的学者认为,算法是程序的灵魂。实际上,对于算法的研究已经有数千年的历史了。计算机的出现,使得用机器自动解题的梦想成为现实。人们可以将算法编成程序交给计算机执行,使得许多原来认为不可能完成的算法变得实际可行。

提示 选择对解题方案进行详细验证,为解题时,减少解题问题的操作步数。

学习效果自评

学完每章的知识后,考生可通过“课后总复习”对所学知识进行检验,还可以对照“学习效果自评”对自己的掌握情况进行检查。

学习效果自评

本章介绍了数据结构与算法的一些相关的概念,重点讲解了算法、数据结构、栈、队列的概念与性质,二叉树的遍历,这些都是以后学习的重点,对于书中的大部分概念只要做到理解就可以了。

考核内容	重要程度	掌握程度	自评结果
算法	★★★★	熟练算法的概念,对两重循环和四重循环的嵌套	<input type="checkbox"/> 优秀 <input type="checkbox"/> 一般 <input type="checkbox"/> 没问题
数据结构	★★★	熟练数据结构的定义,分清线性与非线性结构	<input type="checkbox"/> 优秀 <input type="checkbox"/> 一般 <input type="checkbox"/> 没问题
栈及队列的存储结构	★★	了解栈和队列的基本概念	<input type="checkbox"/> 优秀 <input type="checkbox"/> 一般 <input type="checkbox"/> 没问题
树	★★★★	理解树的概念和术语,掌握树的遍历	<input type="checkbox"/> 优秀 <input type="checkbox"/> 一般 <input type="checkbox"/> 没问题

2. 如何使用书中栏目

书中设计了4个小栏目,分别为“学习提示”、“请注意”、“请思考”和“网络课堂”。

(1) 学习提示

学习提示是从对应模块提炼的重点内容,读者可以通过它明确本部分内容的学习重点和掌握程度。

(2) 请注意

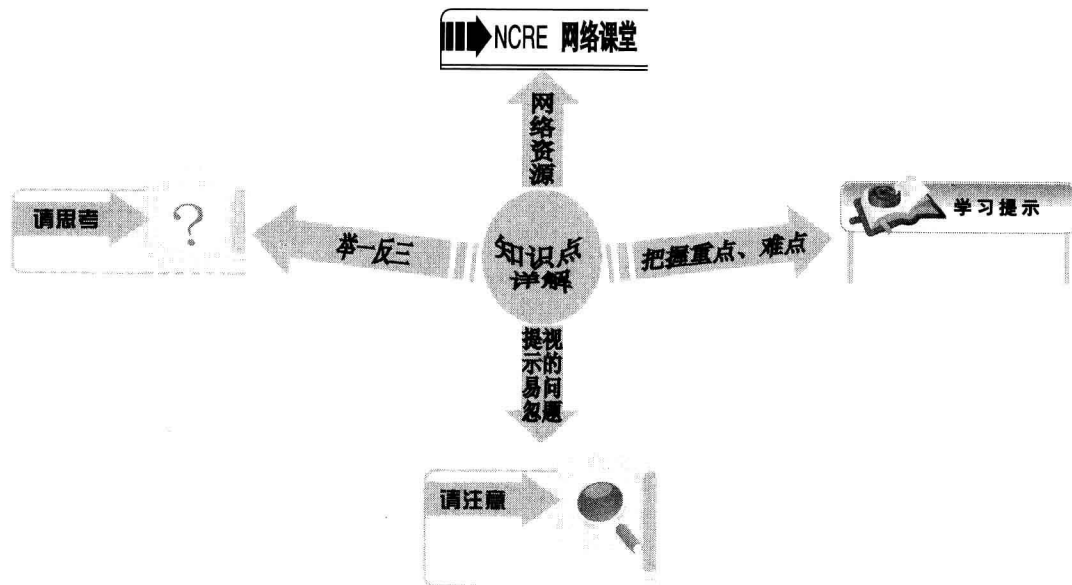
该栏目主要是提示读者在学习过程中容易忽视的问题,以引起大家的重视。

(3) 请思考

介绍完一部分内容后,以这种形式给出一些问题让读者思考,使读者能做到举一反三。

(4) 网络课堂

提供相关扩展知识的网址链接,读者可以通过它们学习更多的知识。



希望本书在备考过程中能够助您一臂之力,让您顺利通过考试,成为一名合格的计算机应用人才。由于时间仓促,书中难免存在疏漏之处,恳请广大读者批评指正。编辑信箱为: lisha@ptpress.com.cn。

编者
2008年11月

多媒体教学光盘使用说明

一、光盘内容

本软件提供多媒体课堂，读者安装本软件后即可使用。

二、光盘使用环境

硬件环境

主 机	Pentium III 相当或以上
内 存	128MB 以上 (含 128MB)
显 卡	SVGA 彩显
硬盘空间	500MB 以上 (含 500MB)

软件环境

操作系统	Windows 2000/XP/、Windows Server 2003
考核形式	选择题前 10 题，填空题前 5 题

三、光盘安装方法

步骤 1: 启动计算机，进入 Windows 操作系统。

步骤 2: 将光盘放入光驱，光盘会自动运行安装程序（也可以双击执行光盘根目录下的 Autorun.exe 文件），将本软件安装到本地硬盘。安装完毕后，会自动在桌面上生成名为“教程二级公共基础知识”的快捷方式。

四、光盘使用方法

1. 启动方法

双击计算机桌面上的“教程二级公共基础知识”快捷方式，弹出如图 1 所示的窗口。

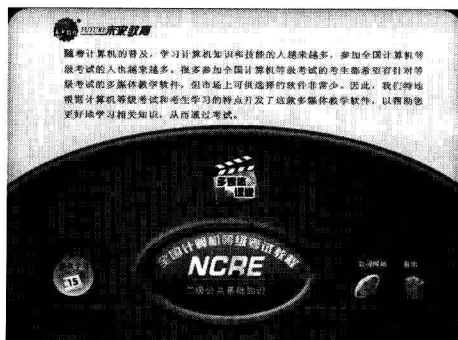


图 1

2. “多媒体课堂”的使用方法

单击图1中的“多媒体课堂”按钮进入多媒体教学课堂，进行交互式学习，如图2所示。

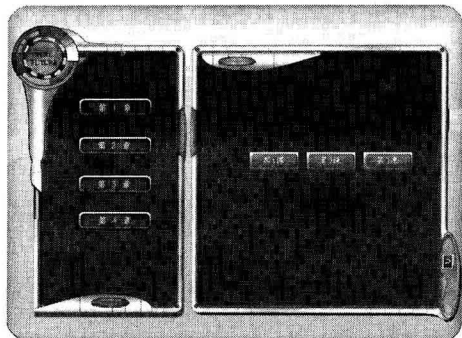


图2

在图2中，单击要学习的章的相应按钮，在界面的右边就会出现该章中对应的课程，然后单击相应课程的按钮即可进入动画学习界面，如图3和图4所示。

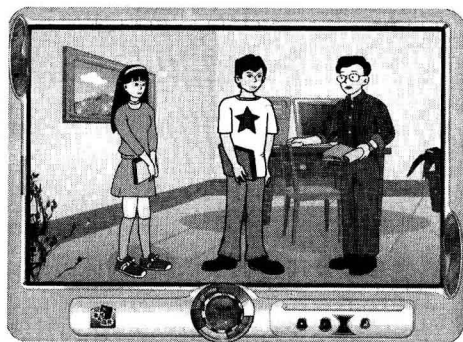


图3

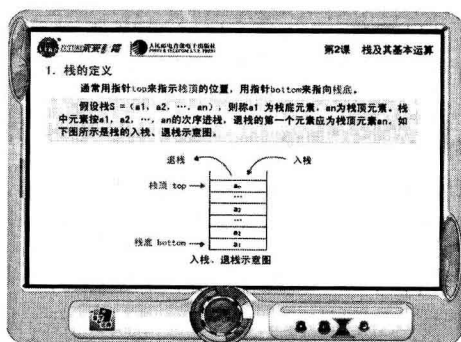


图4

目 录

第1章 数据结构与算法	1	1.8.4 排序方法比较	34
1.1 算法	3	课后总复习	35
1.1.1 什么是算法	3	第2章 程序设计基础	39
1.1.2 算法复杂度	5	2.1 程序设计方法与风格	41
1.2 数据结构的基本概念	6	2.2 结构化程序设计	42
1.2.1 什么是数据结构	7	2.2.1 结构化程序设计方法的重要原则	42
1.2.2 数据结构的图形表示	9	2.2.2 结构化程序的基本结构与特点	43
1.2.3 线性结构与非线性结构	10	2.2.3 结构化程序设计的注意事项	45
1.3 线性表及其顺序存储结构	10	2.3 面向对象的程序设计	45
1.3.1 线性表的基本概念	10	2.3.1 面向对象方法的基本概念	46
1.3.2 线性表的顺序存储结构	11	2.3.2 面向对象方法的优点	49
1.3.3 线性表的插入运算	12	课后总复习	49
1.3.4 线性表的删除运算	13	第3章 软件工程基础	51
1.4 栈和队列	14	3.1 软件工程基本概念	53
1.4.1 栈及其基本运算	14	3.1.1 软件的定义及软件的特点	53
1.4.2 队列及其基本运算	16	3.1.2 软件危机	54
1.5 线性链表	18	3.1.3 软件工程	55
1.5.1 线性链表的基本概念	18	3.1.4 软件工程过程	56
1.5.2 线性链表的基本运算	20	3.1.5 软件生命周期	56
1.5.3 循环链表及其基本运算	22	3.1.6 软件开发工具与开发环境	57
1.6 树与二叉树	22	3.2 结构化分析方法	57
1.6.1 树的基本概念	22	3.2.1 需求分析	58
1.6.2 二叉树及其基本性质	23	3.2.2 需求分析方法	58
1.6.3 二叉树的存储结构	26	3.2.3 结构化分析方法的常用工具	59
1.6.4 二叉树的遍历	26	3.2.4 软件需求规格说明书	60
1.7 查找技术	28	3.3 结构化设计方法	61
1.7.1 顺序查找	28	3.3.1 软件设计概述	61
1.7.2 二分法查找	28	3.3.2 概要设计	62
1.8 排序技术	29	3.3.3 详细设计	65
1.8.1 交换类排序法	29	3.4 软件测试	67
1.8.2 插入类排序法	31		
1.8.3 选择类排序法	33		

3.4.1 软件测试的目的和准则	67	4.2.4 网状模型	89
3.4.2 软件测试方法	68	4.2.5 关系模型	90
3.4.3 白盒测试的测试用例设计	69	4.3 关系代数	92
3.4.4 黑盒测试的测试用例设计	71	4.3.1 关系代数的基本运算	92
3.4.5 软件测试的实施	71	4.3.2 关系代数的扩充运算	94
3.5 程序的调试	73	4.3.3 关系代数的应用实例	96
3.5.1 程序调试的基本概念	73	4.4 数据库设计与管理	96
3.5.2 软件调试方法	74	4.4.1 数据库设计概述	97
课后总复习	75	4.4.2 需求分析	97
		4.4.3 概念设计	98
第4章 数据库设计基础	77	4.4.4 逻辑设计	100
4.1 数据库系统的基本概念	79	4.4.5 物理设计	101
4.1.1 数据库、数据库管理系统、数据库系统	79	4.4.6 数据库管理	101
4.1.2 数据库技术的发展	81	课后总复习	102
4.1.3 数据库系统的基本特点	82		
4.1.4 数据库系统体系结构	83	附录	105
4.2 数据模型	85	附录A 全国计算机等级考试二级公共基础	
4.2.1 数据模型的基本概念	85	知识考试大纲	105
4.2.2 E-R模型	86	附录B 参考答案	106
4.2.3 层次模型	89		

第1章

数据结构与算法



视频课堂

第1课	什么是数据结构 <ul style="list-style-type: none">● 数据的逻辑结构● 数据的存储结构	第2课	栈及其基本运算 <ul style="list-style-type: none">● 栈的定义● 栈的特点● 栈的基本运算
第3课	二叉树及其基本性质 <ul style="list-style-type: none">● 二叉树的定义● 满二叉树和完全二叉树● 二叉树的基本性质● 二叉树的遍历		

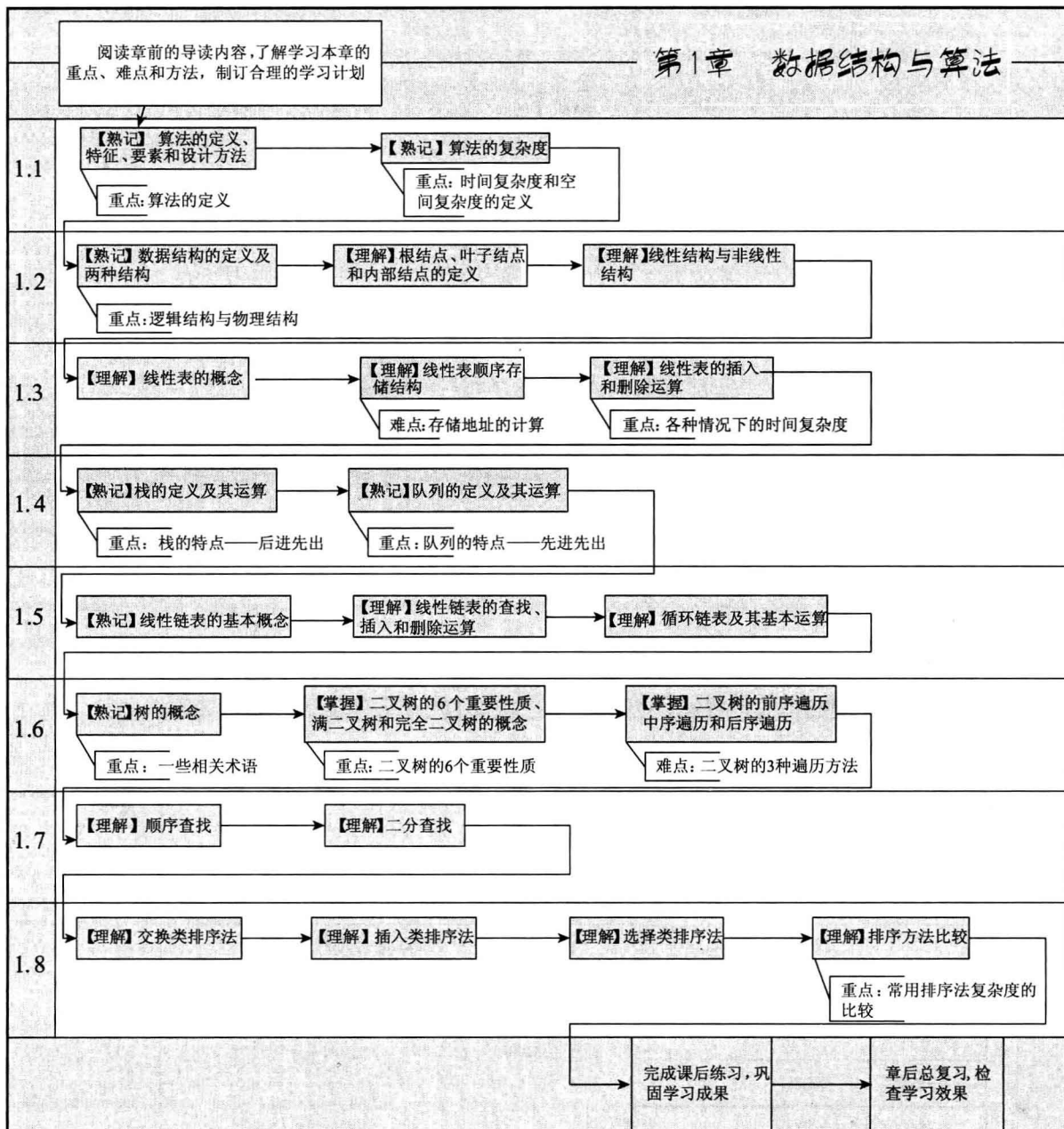
章前导读

通过本章，你可以学习到：

- ◎ 什么是算法？它包含哪些复杂度
- ◎ 什么是数据的逻辑结构和存储结构
- ◎ 栈和队列的定义是什么
- ◎ 二叉树的定义是什么？有哪些性质？二叉树是如何遍历的

本章评估		学习点拨
重要度	★★★★	本章主要介绍算法与数据结构的基础知识。读者在学习的过程中要通过对相关概念的对比理解它们之间的区别和联系。
知识类型	理论	
考核类型	笔试	
所占分值	约10分	
学习时间	10课时	

本章学习流程图



1.1 算 法

本节从算法的基本概念展开,阐述算法的基本特征、基本要素、设计方法以及设计准则,进而详细讲解算法的时间复杂度和空间复杂度。

1.1.1 什么是算法

1. 算法的定义

有的学者认为,算法是程序的灵魂。实际上,对于算法的研究已经有数千年的历史了。计算机的出现,使得用机器自动解题的梦想成为现实,人们可以将算法编写成程序交给计算机执行,使许多原来认为不可能完成的算法变得实际可行。

算法 是指对解题方案的准确而完整的描述,简单地说,就是解决问题的操作步骤。

值得注意的是,算法不等于数学上的计算方法,也不等于程序。在用计算机解决实际问题时,往往先设计算法,用某种表达方式(如流程图)描述,然后再用具体的程序设计语言描述此算法(即编程)。在编程时由于要受到计算机系统运行环境的限制,因此,程序的编制通常不可能优于算法的设计。

2. 算法的基本特征

(1) 可行性

算法在特定的执行环境中执行应当能够得出满意的结果,即必须有一个或多个输出。一个算法,即使在数学理论上是正确的,但如果在实际的计算工具上不能执行,则该算法也是不具有可行性的。

例如,在进行数值计算时,如果某计算工具具有7位有效数字(如程序设计语言中的单精度运算),则在计算下列3个量的和时,

$$A=10^{12}, B=1, C=-10^{12}$$

如果采用不同的运算顺序,就会得到不同的结果,例如:

$$A+B+C=10^{12}+1+(-10^{12})=0$$

$$A+C+B=10^{12}+(-10^{12})+1=1$$

而在数学上, $A+B+C$ 与 $A+C+B$ 是完全等价的。因此,算法与计算公式是有差别的。在设计一个算法时,必须考虑它的可行性。

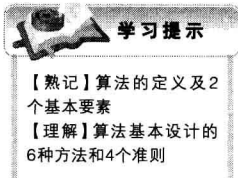
(2) 确定性

算法的确定性表现在对算法中每一步的描述都是明确的,没有多义性,只要输入相同,初始状态相同,则无论执行多少遍,所得的结果都应该相同。如果算法的某个步骤有多义性,则该算法将无法执行。

例如,在进行汉字读音辨认时,汉字“解”在“解放”中读作jie,但它作为姓氏时却读作xie,这就是多义性,如果算法中存在多义性,计算机将无法正确地执行。

(3) 有穷性

算法中的操作步骤为有限个,且每个步骤都能在有限时间内完成。这包括合理的执行时间的含义,如果一个算法执行耗费的时间太长,即使最终得出了正确结果,也是没有意义的。



例如，数学中的无穷级数，当 n 趋向于无穷大时，求 $2n \times n!$ ，显然，这是无终止的计算，这样的算法是没有意义的。

(4) 拥有足够的情报

一般来说，算法在拥有足够的输入信息和初始化信息时，才是有效的；当提供的情报不够时，算法可能无效。

例如， $A=3, B=5$ ，求 $A+B+C$ 的值，显然由于对 C 没有进行初始化，无法计算出正确的答案，所以算法在拥有足够的输入信息和初始化信息时，才是有效的。

在特殊情况下，算法也可以没有输入。因此，一个算法有0个或多个输入。

总之，算法是一个动态的概念，是指一组严谨地定义运算顺序或操作步骤的规则，并且每一个规则都是有效的、明确的，此顺序将在有限的次数下终止。

3. 算法的基本要素

算法的功能取决于两方面因素：选用的操作和各个操作之间的顺序。因此，一个算法通常由两种基本要素组成：

- 对数据对象的运算和操作；
- 算法的控制结构，即运算或操作间的顺序。

(1) 算法中对数据对象的运算和操作

前面介绍了算法的一般定义和基本特征。实际上讨论的算法，主要是指计算机算法。在计算机上可以直接执行的基本操作通常都是用指令来描述的，每个指令代表一种或几种操作。

指令系统	一个计算机系统能执行的所有指令的集合，称为该计算机的指令系统。
------	---------------------------------

指令系统是软件与硬件分界的一个主要标志，是软件与硬件之间相互沟通的桥梁。指令系统在计算机系统中的地位见图1-1。

算法就是按解题要求从指令系统中选择合适的指令组成的指令序列。因此，计算机算法就是计算机能执行的操作所组成的指令序列。不同计算机系统，指令系统是有差异的，但一般的计算机系统中，都包括以下4类基本的运算和操作，如表1-1所示。

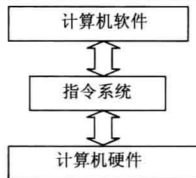


图1-1 计算机的体系结构

表1-1 4类基本的运算和操作

运算类型	操作	例子
算术运算	+、-、×、÷	$a+b, 3-1\dots$
逻辑运算	与(&)、或()、非(!)	$!1, 1 0, 1&1\dots$
关系运算	>、<、=、≠	$a>b, a=c, b\neq c\dots$
数据传输	赋值、输入、输出	$a=0, b=3\dots$

(2) 算法的控制结构

算法的控制结构是算法中各个操作之间的执行顺序。

算法一般是由顺序、选择（又称分支）和循环（又称重复）3种基本结构组合而成。

描述算法的工具传统的流程图、N-S结构化流程图和算法描述语言等。

图1-2所示是用流程图方式表示的选择结构的两种类型。

图1-2(a)的执行步骤如下所述。

- ◆ 步骤1 X 赋值为2。
- ◆ 步骤2 判断 X 的值是否小于3，条件成立。
- ◆ 步骤3 X 的值减少2。

● 步骤4 输出 X 的值,最后结果为0。

图1-2(b)的执行步骤如下所述。

● 步骤1 X 赋值为2。

● 步骤2 X 的值增加2。

● 步骤3 判断 X 的值是否小于3,条件不成立。

● 步骤4 输出 X 的值,最后结果为4。

图1-2(a)执行的是先判断 X 的值是否小于3,如果条件成立则 X 的值减2,最终结果为0,而图1-2(b)先将 X 的值增加2,然后再判断 X 的值是否小于3,最终结果为4。

从中可以看出,选用的基本操作虽然相同,但由于存在执行顺序的差异,得到的结果却完全不同。

4. 算法基本设计方法

虽然设计算法是一件非常困难的工作,但是算法设计也不是无章可循的,人们经过实践,总结和积累了许多行之有效的方法。常用的几种算法设计方法有列举法、归纳法、递推法、递归法、减半递推技术和回溯法。

1.1.2 算法复杂度

一个算法的复杂度高低体现在运行该算法所需要的计算机资源的多少,所需的资源越多,就说明该算法的复杂度越高;反之,所需的资源越少,则该算法的复杂度越低。计算机的资源,最重要的是时间和空间(即存储器)资源。

因此,算法复杂度包括算法的时间复杂度和算法的空间复杂度。

1. 算法的时间复杂度

算法的时间复杂度 是指执行算法所需要的计算工作量。

值得注意的是:算法程序执行的具体时间和算法的时间复杂度并不是一致的。算法程序执行的具体时间受到所使用的计算机、程序设计语言以及算法实现过程中的许多细节所影响。而算法的时间复杂度与这些因素无关。

算法的计算工作量是用算法所执行的基本运算次数来度量的,而算法所执行的基本运算次数是问题规模(通常用整数 n 表示)的函数,即

$$\text{算法的工作量} = f(n)$$

其中 n 为问题的规模。

所谓问题的规模就是问题的计算量的大小。如 $1+2$,这是规模比较小的问题,但 $1+2+3+\dots+10000$,这就是规模比较大的问题。

例如,在下列3个程序段中:

① $\{x++;s=0\}$

② $\text{for}(i=1;i \leq n;i++)$

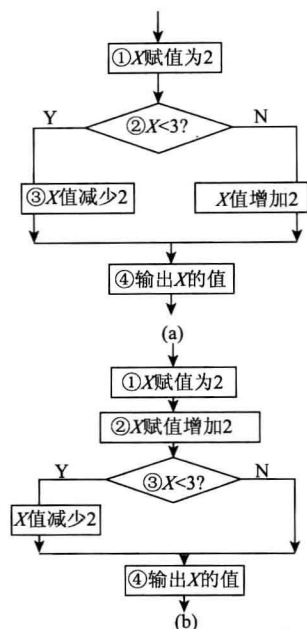


图1-2 算法的控制结构

学习提示

【熟记】算法的时间复杂度和空间复杂度各自的概念及度量

{x++;s+=x;}/*一个简单的for循环,循环体内操作执行了n次*/

③ for (i=1;i<=n;i++)

for (j=1;j<=n;j++)

{x++;s+=x;}/*嵌套的双层for循环,循环体内操作执行了 n^2 次*/

①中,基本运算“x++”只执行一次。重复执行次数分别为1;

②中,由于有一个循环,所以基本运算“x++”执行了n次;

③中,嵌套的双层循环,所以基本运算“x++”执行了 n^2 次。

则这3个程序段的时间复杂度分别为 $O(1)$ 、 $O(n)$ 和 $O(n^2)$ 。

在具体分析一个算法的工作量时,在同一个问题规模下,算法所执行的基本运算次数还可能与特定的输入有关。即输入不同时,算法所执行的基本运算次数不同。例如,使用简单插入排序算法(见本书1.8节),对输入序列进行从小到大排序。输入序列为:

a.1 2 3 4 5

b.1 3 2 5 4

c.5 4 3 2 1

我们不难看出,序列a所需的计算工作量最少,因为它已经是非递减顺序排列,而序列c将耗费的基本运算次数最多,因为它完全是递减顺序排列的。

在这种情况下,可以用以下两种方法来分析算法的工作量。

- 平均性态;
- 最坏情况复杂性。

请思考



算法的复杂度是以什么来度量的?

2 算法的空间复杂度

算法的空间复杂度 是指执行这个算法所需要的内存空间。

算法执行期间所需的存储空间包括3个部分:

- 输入数据所占的存储空间;
- 程序本身所占的存储空间;
- 算法执行过程中所需要的额外空间。

其中,额外空间包括算法程序执行过程中的工作单元,以及某种数据结构所需要的附加存储空间。

如果额外空间量相对于问题规模(即输入数据所占的存储空间)来说是常数,即额外空间量不随问题规模的变化而变化,则称该算法是原地(in place)工作的。

为了降低算法的空间复杂度,主要应减少输入数据所占的存储空间以及额外空间,通常采用压缩存储技术。

1.2 数据结构的基本概念

在进行数据处理时,实际需要处理的数据一般有很多,而这些大量的数据都需要存放在计算机中,因此,大量的数据在计算机中如何组织,才能提高数据处理的效率,节省计算机的存储空间呢?

通过本节的学习,可以了解什么是数据结构,它们是如何用图形表示的,以及线性结构与非线性结构的区别。

1.2.1 什么是数据结构

数据结构研究的内容包括3个方面:

- 数据集中各数据元素之间所固有的逻辑关系,即数据的逻辑结构;
- 在对数据进行处理时,各数据元素在计算机中的存储关系,即数据的存储结构;
- 对各种数据结构进行的运算。

其中,数据元素是一个含义很广泛的概念。它是数据的“基本单位”,在计算机中通常作为一个整体进行考虑和处理。在数据处理领域中,每一个需要处理的对象,甚至于客观事物的一切个体,都可以抽象成数据元素,简称为元素。

例如:

- 日常生活中一日三餐的名称——早餐、午餐、晚餐,可以作为一日三餐的数据元素;
- 在地理学中表示方向的方向名称——东、南、西、北,可以作为方向的数据元素;
- 在军队中表示军职的名称——连长、排长、班长、战士,可以作为军职的数据元素。

如果要给数据结构(Data Structure)下一个完整而准确的定义,那将是一件非常困难的事情。对数据结构的概念,在不同的书中,有不同的提法,顾名思义,所谓数据结构,包含两个要素,即“数据”和“结构”。

数据	是需要处理的数据元素的集合,一般来说,这些数据元素,具有某个共同的特征。
----	--------------------------------------

例如,东、南、西、北这4个数据元素都有一个共同的特征,它们都是地理方向名,分别表示二维地理空间中的4个方向,这4个数据元素构成了地理方向名的集合。

又例如,早餐、午餐、晚餐这3个数据元素也有一个共同的特征,即它们都是一日三餐的名称,从而构成了一日三餐名的集合。

结构	所谓“结构”,就是关系,是集合中各个数据元素之间存在的某种关系(或联系)。
----	---------------------------------------

“结构”是数据结构研究的重点。数据元素根据其之间的不同特性关系,通常可以分为以下4类:线性结构(图1-3(a))、树形结构(图1-3(b))、网状结构(图1-3(c))和集合(图1-3(d))。

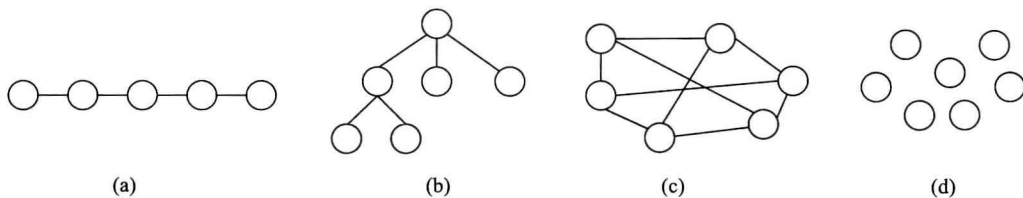


图1-3 4类基本结构

在数据处理领域中,通常把两两数据元素之间的关系用前后件关系(或直接前驱与直接后继关系)来描述。实际上,数据元素之间的任何关系都可以用前后件关系来描述。

例如,在考虑一日三餐的时间顺序关系时,“早餐”是“午餐”的前件(或直接前驱),而“午餐”是“早餐”的后件(或直接后继);同样,“午餐”是“晚餐”的前件,“晚餐”是“午餐”的后件。

又例如,在考虑军队中的上下级关系时,“连长”是“排长”的前件,“排长”是“连长”的后件,“排长”是“班



学习提示

【熟记】数据结构研究的3方面内容、数据逻辑结构与物理结构的概念