

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

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，丰富你的学习方式
- 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



NORIE

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



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

人民邮电出版社

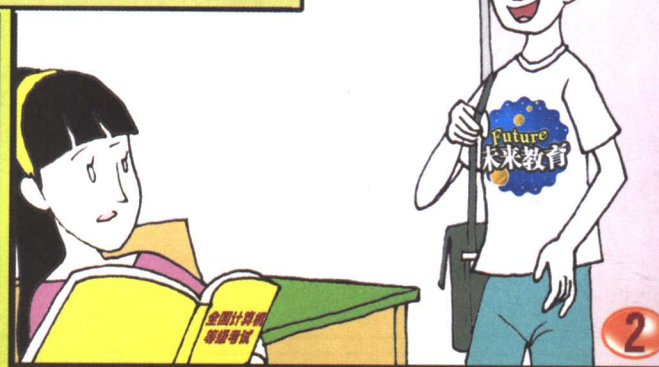
北京

小精灵无忧备考记



“考试何其难，复习没时间”，有谁能来帮帮我呀！

1



怎么了，小精灵？一副愁眉苦脸的样子。说来听听，看我能不能帮助你。

2



最近报考了计算机等级考试，本来信心十足，可是……

3



……我正要上课呢，我带你去见考试通老师吧。

4



老师，您好！这是小精灵，她也报考了计算机等级考试，有些问题想向您请教。

小精灵，你好！学习中遇到问题是很正常的，你也跟电脑迷一起来上课吧。

老师，您好！

5

全国计算机等级考试培训

0 课前导读

通过本章，你可以学习到：

- 计算机语言的分类
- 算法的基本概念及特点
- 结构化程序设计的基本概念
- Turbo C语言

本章学习效果自评

学完本章后，相信大家程序设计方面的知识有了一定的了解，本章内容在考试中多以选择题方式出现，下面是我们对本章学习效果进行的自评，大家可以对照自己的知识掌握情况进行判断。

考核内容	自评	自评	自评
理解程序语言的基本概念	★	□	□
理解算法的基本概念	★★★	□	□
理解结构化程序设计的基本概念	★★	□	□
理解 Turbo C 语言	★	□	□
理解程序语言的基本概念	★★★	□	□
理解算法的基本概念	★★	□	□

学习难点	本章评估
学习难点	★
知识类型	概述
考核类型	笔试
所占分值	20分-40分
学习时间	2课时

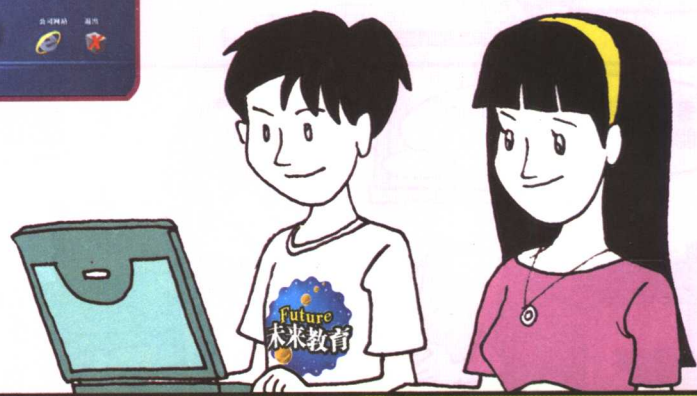
学习中要抓住重点，掌握学习的技巧，把握知识脉络，你们要仔细阅读书中相应的栏目。

6

随着计算机的普及，选择计算机进行学习的人越来越多，而且参加考试的人也越来越多。为了更好地帮助全国计算机等级考试的考生能够更有效地掌握考试的多媒体教学软件，特编、出版全国计算机等级考试教材的配套，即配有多媒体教学软件的磁带、光盘。虽然出版全国计算机等级考试教材的配套，但配有多媒体教学软件的磁带、光盘。我们开发了这款具有代表性的全国计算机等级考试教材多媒体教学软件。



在学习的过程中，你们还可以结合本书所配的多媒体教学光盘对某些知识点进行巩固。



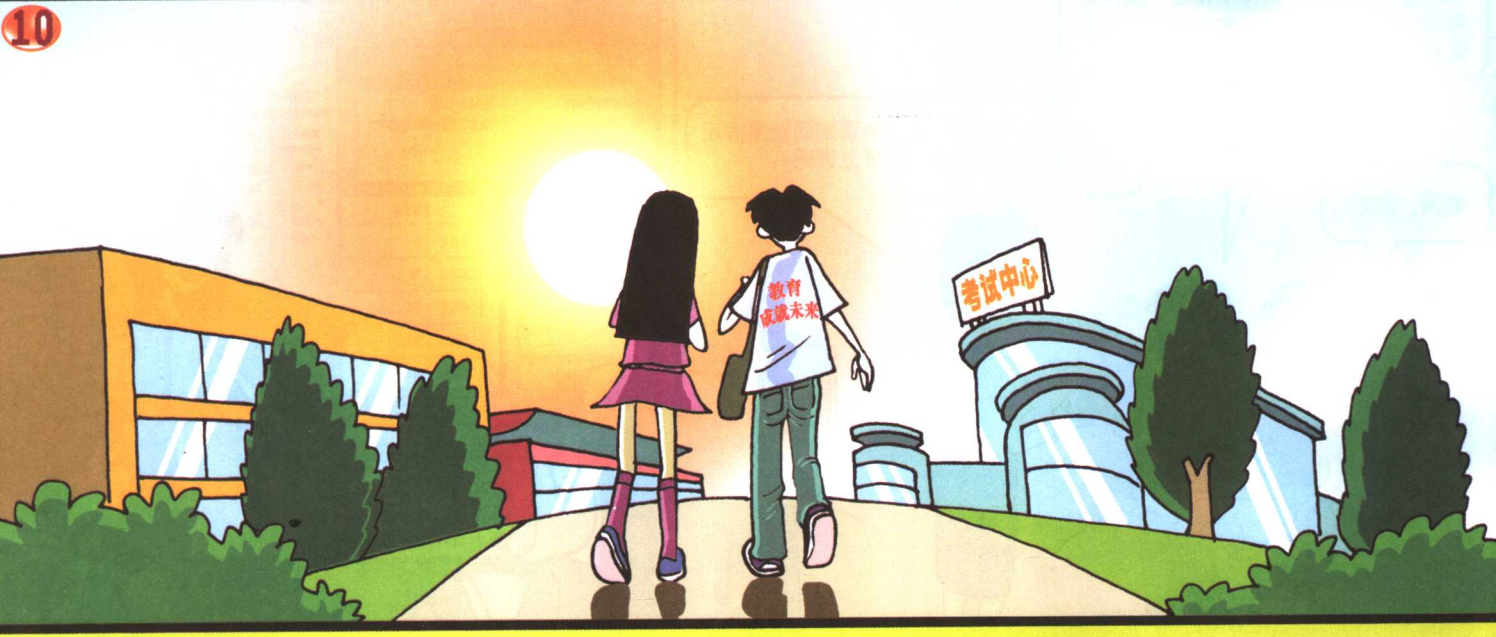
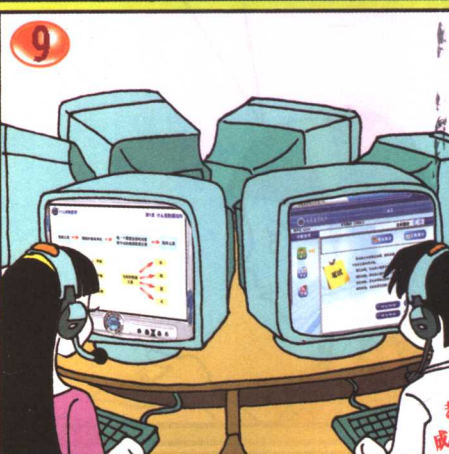
可是现在有很多软件在使用的时候都比较麻烦，谁来教我们如何使用呢。

到今天为止，所有的课程已讲完了，下面应该多做一些练习。最后，建议你们通过软件中的模拟考试对自己的学习效果进行自评。

书上有光盘使用说明，而且还写得很详细，仔细阅读肯定能应用自如。



全国计算机等级考试培训



编 委 会

主 编：张 淼

委 员（按姓氏笔画排列，排名不分先后）：

付红伟 任 威 李 琴 谷永生 张 涛

张 萍 张 琦 张 燕 张冬梅 张圣亮

侯 军 祝 萍 咎 超 郑慧芳 钱 勇

唐彦文 梁敏勇

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、为什么编写本书

计算机等级考试的时间周期短,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,并且大多考生是非计算机专业的学生或社会人员,基础比较薄弱,计算机知识掌握得不够牢固,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(尤其是初学者类考生)学习困难的方法,可提高学习效率和学习效果。因此我们策划出版了本套丛书,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

本套丛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。在学习各章节的内容之前,都详细介绍了考试中的重点和难点,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

本套丛书所配的光盘是全新的视频教学光盘,用动画演绎复杂的理论知识,用视频讲解各种操作方法。使用本套丛书的光盘,就等于把辅导老师请回了家。

- 一一点拨重点、难点,一步步深入学习;运用动画、视频等多媒体技术,让学习变得轻松而高效。
- 熟悉上机考试的环境;享誉多年的上机模拟软件,让你提前感受上机考试的全过程。

三、如何学习本套丛书

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

学习点拨

提示每章知识点中的重点和难点,为考生介绍学习方法,使考生更有针对性地学习。

学习点拨

本章我们要讲到的都是一些算法、数据结构的基础知识,在学习的过程中要通过对相关概念的相互对照理解它们之间的区别和联系,从而达到对知识的掌握。

本章评估

通过对数套历年笔试题目的分析,总结出每章的重要程度、考核类型、所占分值,以及建议学习时间等重要参数,使读者可以更加合理地安排学习。

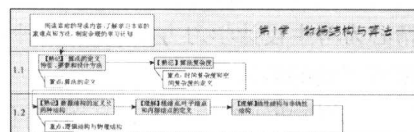
本章评估

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,同时指出每一个知识点的掌握程度: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,使读者迅速把握重点,轻松获得知识,最终顺利通过考试。

1.1 算 法

本章从算法的基本概念讲起,阐述算法的基本特征、算法要素、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,达到了解算法的目的。

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存在数千年的历史了。计算机的发明,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,使许多原来认为不可能完成的算法变得实际可行。

算法 既能写解问题的逻辑性思维,又能表达,就是解决问题的具体步骤。



本章学习效果自评

提供“复习题”供考生对所学知识进行检验,最后考生还可以对照“学习效果自评表”对本章知识的掌握情况进行一次自我检查。

本章学习效果自评

本章我们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、二叉树的概念与性质。二叉树的遍历,这些都是以后学习的基础,对于书中的大部分概念只要能够理解就可以了。

学习内容	重要程度	掌握情况	自评结果
算法	★★★★	熟悉算法的概念、时间复杂度和空间复杂度的概念	<input type="checkbox"/> 不懂 <input type="checkbox"/> 一般 <input type="checkbox"/> 较好
数据结构	★★★★	熟悉数据结构的定义、分类、特点并理解逻辑与存储结构的区别	<input type="checkbox"/> 不懂 <input type="checkbox"/> 一般 <input type="checkbox"/> 较好
线性表及其顺序存储结构	★	了解线性表的由来概念	<input type="checkbox"/> 不懂 <input type="checkbox"/> 一般 <input type="checkbox"/> 较好
栈	★★★★	理解栈的概念和特点,掌握栈的运算	<input type="checkbox"/> 不懂 <input type="checkbox"/> 一般 <input type="checkbox"/> 较好

2. 如何使用本书栏目

本书设计了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

我们以这种形式提示考生在学习的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进行思考,从而做到举一反三。

(4) 网络课堂

提供相关扩展知识的网址链接,读者可以利用它们学习更多的知识。



本书对于参加计算机等级考试的考生,是一本极其实用的教材。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: jiangyan@ptpress.com.cn。

编者
2006年12月

多媒体教学光盘使用说明

一、光盘内容

本系统主要包括大纲知识点的多媒体课堂，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 II 以上及其各种兼容机
内 存	64MB 以上的基本内存
显 卡	彩色显示器，具备 512KB 以上显存
硬盘空间	100MB 以上的剩余空间

软件环境

操作系统	Windows 2000/XP/2003
考试系统	全国计算机等级考试系统
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，进行互动式学习环境，如图2所示。

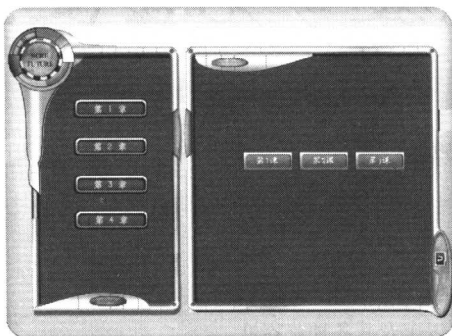


图2

在图2中，单击要学习的章，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

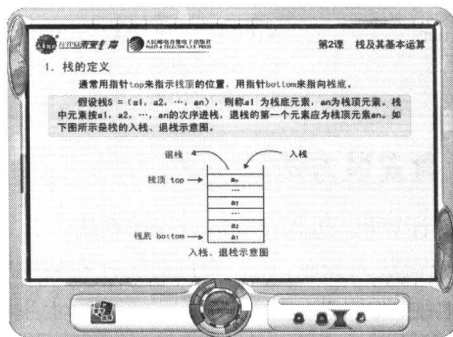


图4

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	43
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	46
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 顺序表的插入运算	12	课后总复习	49
1.3.4 顺序表的删除运算	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	4.2.2 E-R模型	86
3.4.2 软件测试方法	68	4.2.3 层次模型	89
3.4.3 白盒测试的测试用例设计	69	4.2.4 网状模型	89
3.4.4 黑盒测试的测试用例设计	71	4.2.5 关系模型	90
3.4.5 软件测试的实施	71	4.3 关系代数	92
3.5 程序的调试	73	4.3.1 关系代数的基本运算	92
3.5.1 程序调试的基本概念	73	4.3.2 关系代数的扩充运算	94
3.5.2 软件调试方法	74	4.3.3 关系代数的应用实例	96
课后总复习	75	4.4 数据库设计与管理	96
第4章 数据库设计基础	77	4.4.1 数据库设计概述	97
4.1 数据库系统的基本概念	79	4.4.2 需求分析	97
4.1.1 数据库、数据库管理系统、数据库系统	79	4.4.3 概念设计	98
4.1.2 数据库技术的发展	81	4.4.4 逻辑设计	100
4.1.3 数据库系统的基本特点	82	4.4.5 物理设计	101
4.1.4 数据库系统体系结构	83	4.4.6 数据库管理	101
4.2 数据模型	85	课后总复习	102
4.2.1 数据模型的基本概念	85	附录 参考答案	105

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		

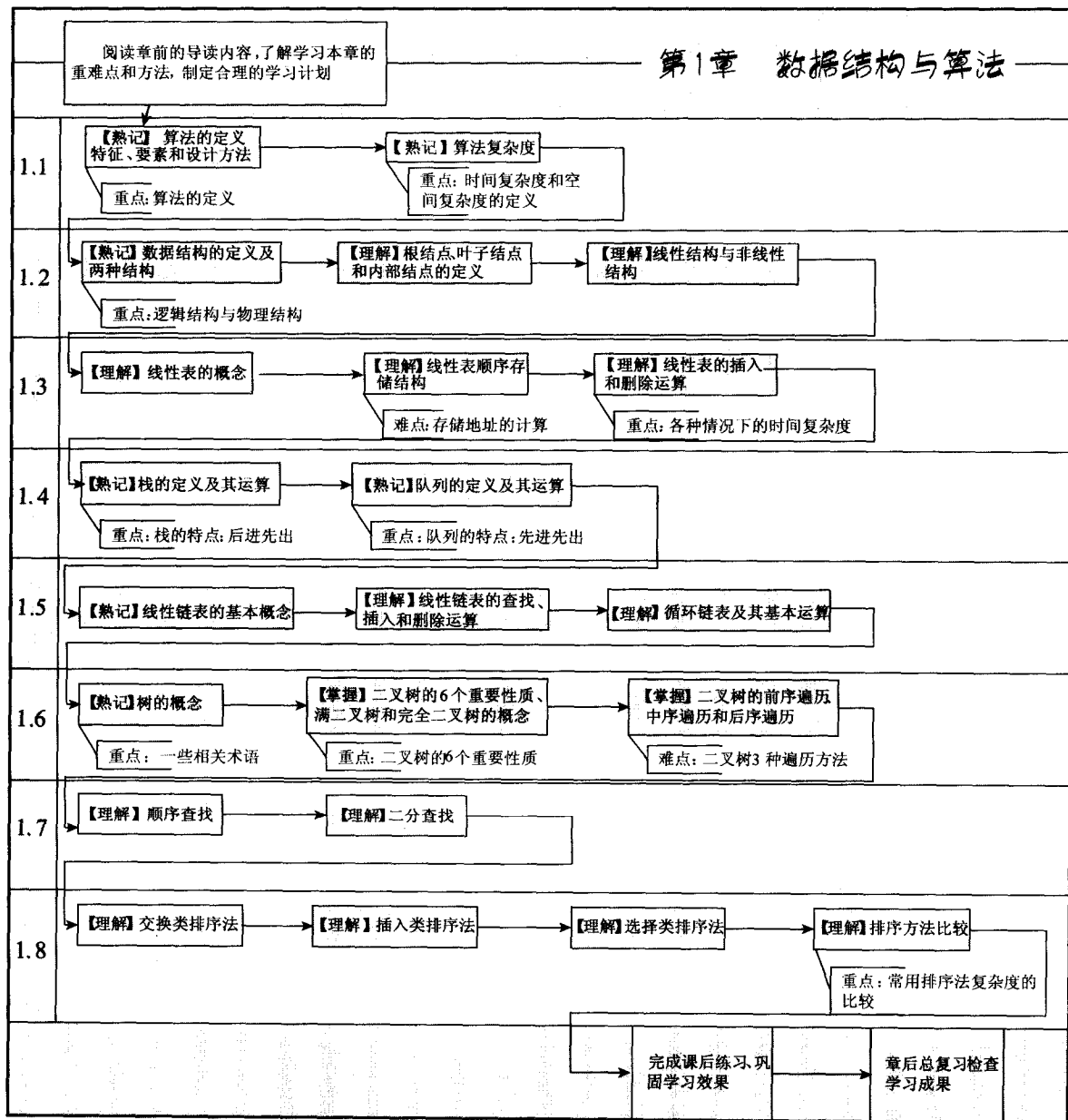
章前导读

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

本章评估		学习点拨
重要度	★★★★	本章我们要讲到的都是一些算法、数据结构的基础知识，在学习的过程中要通过对相关概念的相互对照理解它们之间的区别和联系，从而达到对知识的掌握。
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,达到了解算法的目的。

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,简单的说,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。



学习提示

【熟记】算法的定义及2个基本要素

【理解】算法基本设计的6种方法和4个准则

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面的一个主要标志, 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所示。

算法就是按算法要求从指令系统中选择合适的指令, 组成指令序列。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。

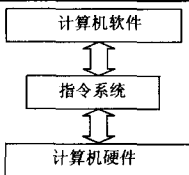


图1-1 计算机的体系结构

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)、循环(又称重复)3种基本结构组合而成。

描述算法的工具具有传统的流程图、N-S结构化流程图、算法描述语言等。

图1-2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，人们经过几十年来的实践，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。

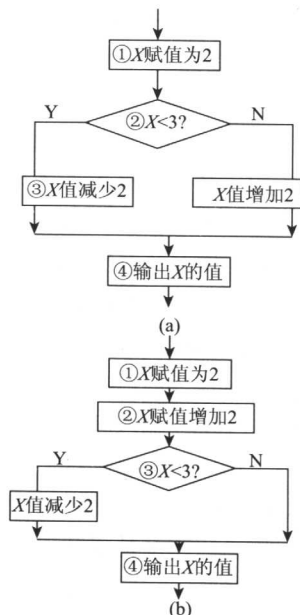


图1-2 算法的控制结构

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计算工作量并不是一致的。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+...+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 for (i=1;i<=n;i++)

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