

★依据教育部最新大纲及教材编写



天一文化
TIANYI CULTURE

全国计算机 等级考试

指定教程名师辅导 二级公共基础

全国计算机等级考试指定教程名师辅导编委会 编



北京邮电大学出版社

全国计算机等级考试指定教材名师辅导

二级公共基础

全国计算机等级考试指定教材名师辅导编写组 编

北京邮电大学出版社

· 北京 ·

内 容 简 介

本书是根据教育部考试中心 2004 年新制定的《全国计算机等级考试大纲》中关于二级公共基础部分的要求编写的。书中各章的基本要求是对大纲的细化,简明扼要地叙述了大纲中对数据结构与算法、程序设计基础、软件工程基础以及数据库设计基础的基本要求。大量的例题分析与综合练习,完全可以满足考生应考的需要。

本书不仅可以作为应试者的辅导教材,也可以作为高等院校相应课程的参考书。

图书在版编目(CIP)数据

二级公共基础/《全国计算机等级考试指定教材名师辅导》编写组编.

—北京:北京邮电大学出版社,2004

ISBN 7-5635-0956-9

I. 二… II. 全… 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(2004)第 104238 号

书 名:二级公共基础

编 著:全国计算机等级考试指定教材名师辅导编写组

责任编辑:李茂林

出版发行:北京邮电大学出版社

社 址:北京市海淀区西土城路 10 号(邮编:100876)

电话传真:010—62282185(发行部) 010—62283578(Fax)

电子邮箱:publish@bupt.edu.cn

经 销:各地新华书店

印 刷:北京恒达印刷有限公司

开 本:787mm×1 092mm 1/16

印 张:8.25

字 数:170 千字

印 数:1~5000 册

版 次:2004 年 12 月第 1 版

ISBN 7-5635-0956-9/TP·129

定价:16.00 元

如有质量问题请与发行部联系

版权所有 侵权必究

全国计算机等级考试系列辅导用书

编委会主任委员会

主 任

刘振安：中国科学技术大学教授、计算机类教材编审委员会副主任、全国计算机等级考试委员会委员、全国高等教育自学考试委员会委员

副主任(按姓氏笔画排列)

卢炎生：华中科技大学教授、全国计算机等级考试委员会委员、全国高等教育自学考试委员会委员

李明星：哈尔滨工业大学教授、全国高等教育自学考试委员会委员

杨 洪：山东大学教授、全国计算机等级考试委员会委员

张西怀：西安交通大学教授、全国高等教育自学考试委员会委员

陈怀义：国防科技大学教授、全国自考委电子电工类专业委员会副主任

侯文永：上海交通大学教授、全国高等教育自学考试委员会委员

侯炳辉：清华大学教授、全国高等教育自学考试委员会委员

高传善：复旦大学教授、全国计算机等级考试委员会委员

出版说明

全国计算机等级考试自开考以来,得到了社会的广泛认可,考试人数逐年增加,至2004年底,参加考试人数累计超过1 350万人,其中400多万人获得了等级证书。计算机等级考试证书已成为许多单位人员录用、任职和晋升的重要条件之一。

为适应计算机应用技术的发展,教育部考试中心对全国计算机等级考试的开考科目、考核内容及考试形式均进行了较大调整,推出了2004年版考试大纲。调整后的全国计算机等级考试分为四个等级:

一级 包括:一级 MS Office

一级 WPS Office

一级 B

考生可任选一种参加考试。

二级 包括:二级 C 语言程序设计

二级 C ++ 语言程序设计

二级 Java 语言程序设计

二级 Visual Basic 语言程序设计

二级 Visual Foxpro 程序设计

二级 Access 数据库程序设计

考生可任选一种参加考试。

C ++、Java、Access 是本次新开考的内容,同时增加了二级公共基础知识的考核。

提示:新大纲增加的二级公共基础知识并不以单独的试卷进行考试,而是把这部分的内容加在每种语言的考试试卷中,其所占比例为30%。因此,原来开考的二级考试的科目,例如C语言程序设计,其难度也有所提高。

三级 包括:三级 PC 技术

三级信息管理技术

三级网络技术

三级数据库技术

考生可任选一种参加考试。

四级

为帮助考生掌握最新大纲教材内容和通过考试,我们组织编写了这套全国计算机等级考试指定教材名师辅导丛书。

丛书包括两个系列:(1)全国计算机等级考试指定教材名师辅导

(2)全国计算机等级考试标准预测试卷

(3)全国计算机等级考试上机考试真题解析

本套丛书有以下特点:

一、作者阵容强大、熟悉等级考试规律

丛书作者来自全国各重点高校,全部是国内计算机界资深专家和高校一线教师。由全国计算机等级考试委员会委员和全国高等教育自学考试专业委员会委员担任编委会主任和副主任,主任委员会负责制订丛书的写作体例、内容要求和审定书稿。每科的编写则由一位主任委员具体负责,组成编写组,参与编写的人员均对计算机等级考试有深入的研究,有丰富的计算机等级考试辅导评卷经验,熟悉计算机等级考试规律,编写过多本有影响力的计算机等级考试教材和辅导教材。

二、内容科学严谨、针对性强

丛书编写紧扣最新考试大纲,总结了近年来计算机等级考试的规律。每科由学习方法、历年试题规律总结、应试技巧、考核重点、典型及综合例题分析、同步综合练习、参考答案和重点解析、上机考试指导、模拟试卷以及历年试题组成。

编写中遵循少而精的原则,用较少篇幅把必考内容、全真习题、精辟解析等全面覆盖,重点突破考试中的疑点与难点,使考生用较少的时间与精力,便能掌握考试全局的精髓,真正做到胸中有丘壑,笔底出江山。

习题的设计是辅导用书的命脉,判断辅导用书好坏的关键是看各种习题质量的高低。如果习题质量不高,抓不住重点,答案要点不全、不准确,会使考生浪费大量的宝贵时间而不得要领,从而误导考生,贻害无穷。本套丛书习题的设计从新大纲出发,紧扣考点,答案准确、精炼,题型、题量、难易度与考试真题完全一致,有极强的针对性,真正做到有的放矢,考生只要能认真习作,举一反三,考试过关自然轻松自如。

三、传授临场解题经验、轻松过关

本丛书的各个科目都将以一套全真试卷为例,从临场考试的角度示范如何在考场上准确而快捷地进行解答,以便让考生获得宝贵的实战经验,轻松过关。

四、编校严格负责、质量上乘

北京邮电大学出版社是一家以出版计算机、通信类图书闻名的专业出版社,计算机图书编辑力量很强,编校质量上乘。本套丛书经过出版社编辑的严格把关,保证了图书的高质量。

丛书的出版可以说是计算机等级考试辅导用书的一个里程碑,对考生应考会有相当大的帮助,一切为了考生,质量说明一切,相信考生在购买和使用同类辅导用书时会进行认真的比较。

全国计算机等级考试指定教材名师辅导编委会

前 言

本书是计算机等级考试辅导丛书之一。编写本书的根据是教育部颁布的《全国计算机等级考试大纲(2004)》中关于二级公共基础部分的要求。

根据新的计算机等级考试大纲要求,对于二级中的所有科目增加基础知识部分,并且对基础知识统一要求,称为二级公共基础。二级公共基础的内容主要包括数据结构与算法、程序设计基础、软件工程基础、数据库设计基础四个部分。

本书根据二级公共基础中的四部分内容分为四章。每一章均分为基本要点、典型例题分析、同步综合练习和同步综合练习参考答案等四个专题。

在每一章的基本要求这个专题中,简明扼要地叙述了大纲中对这部分内容的基本要求,它实际上是大纲中对这部分内容要求的细化,也反映了考核的基本内容和深度。

在每一章的典型例题分析这个专题中,以例题的方式进一步阐述了本章所要掌握的基本概念以及本章中一些主要的知识点。

在每一章的同步综合练习这个专题中,给出了一定数量的练习,作为对本章主要知识点的补充。

在每一章的最后一个专题中,给出了本章同步综合练习这个专题中各练习的参考答案。

本书的特点是内容精炼,语言通俗易懂。它不仅可以满足全国计算机等级考试(二级公共基础知识)的需要,而且也可以作为一般院校相应课程的自学参考书。

由于时间紧迫,以及作者水平有限,书中难免有错误或不妥之处,恳请读者批评指正。

作 者

2004 年 9 月

目 录

二级公共基础的学习与考试	(1)
1.1 学习方法	(1)
1.2 应试技巧	(2)
第1章 数据结构与算法	(3)
1.1 基本要点	(3)
1.1.1 算法的基本概念	(3)
1.1.2 数据结构的基本概念	(7)
1.1.3 线性表及其顺序存储结构	(10)
1.1.4 栈和队列	(12)
1.1.5 线性链表	(14)
1.1.6 树与二叉树	(20)
1.1.7 查找技术	(24)
1.1.8 排序技术	(25)
1.2 典型例题和综合例题分析	(28)
1.3 同步综合练习	(35)
一、选择题	(35)
二、填空题	(37)
1.4 参考答案及重点解析	(38)
一、选择题	(38)
二、填空题	(39)
第2章 程序设计基础	(41)
2.1 基本要点	(41)
2.1.1 程序设计的方法	(41)
2.1.2 程序设计的风格	(43)
2.1.3 面向对象的程序设计	(45)
2.2 典型例题和综合例题分析	(47)
2.3 同步综合练习	(50)
一、选择题	(50)
二、填空题	(52)
2.4 参考答案及重点解析	(52)
一、选择题	(52)
二、填空题	(53)
第3章 软件工程基础	(54)
3.1 基本要点	(54)

3.1.1 软件工程的基本概念	(54)
3.1.2 结构化分析方法	(62)
3.1.3 结构化设计方法	(65)
3.1.4 测试与调试	(69)
3.2 典型例题和综合例题分析	(75)
3.3 同步综合练习	(80)
一、选择题	(80)
二、填空题	(81)
3.4 参考答案及重点解析	(82)
一、选择题	(82)
二、填空题	(84)
第4章 数据库设计基础	(87)
4.1 基本要点	(87)
4.1.1 数据库的基本概念	(87)
4.1.2 数据描述与数据模型	(90)
4.1.3 关系代数	(94)
4.1.4 数据库设计方法	(100)
4.2 典型例题和综合例题分析	(106)
4.3 同步综合练习	(111)
一、选择题	(111)
二、填空题	(112)
4.4 参考答案及重点解析	(113)
一、选择题	(113)
二、填空题	(115)
附 录	(117)

二级公共基础应试技巧

1.1 学习方法

根据新大纲的要求,二级公共基础中各部分所包括的内容如下。

1. 基本数据结构与算法的内容

- 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)。
- 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。

- 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。
- 栈和队列的定义;栈和队列的顺序存储结构及其基本运算。
- 线性单链表、双向链表与循环链表的结构及其基本运算。
- 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。
- 顺序查找与二分查找算法;基本排序算法(交换类排序,选择类排序,插入类排序)。

2. 程序设计基础的内容

- 程序设计方法与风格;结构化程序设计。
- 面向对象的程序设计方法、对象、方法、属性及继承与多态性。

3. 软件工程基础的内容

- 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。
- 结构化分析方法;数据流图;数据字典;软件需求规格说明书。
- 结构化设计方法;总体设计与详细设计。
- 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。
- 程序的调试;静态调试与动态调试。

4. 数据库设计基础的内容

- 数据库的基本概念;数据库;数据库管理系统;数据库系统。
- 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。
- 关系代数理论中的运算,包括集合运算及选择、投影、连接运算;数据库规范化理论。
- 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。

虽然二级公共基础所包括的内容比较多,但根据考试大纲的要求,每一部分所要求的深度并不很高,只要掌握其中的基本概念就行,并不要求掌握具体的算法以及更深层次的应用。掌握这些基本概念的目的是要求在具体考试科目中能掌握一些简单的应用。因此,考生在学习这部分内容时,要重点掌握基本概念,熟悉和理解其中的基本术语。当然,对于一些基本的方法也要了解,以便在具体考试科目中会使用这些方法。例如,在数据结

构中所涉及的最基本的查找与排序的方法,在二级公共基础的考试中可能只考这些方法的特点以及每种方法的计算工作量,而不涉及具体的算法描述,但这些算法可能要在程序设计语言的考试中出现,考生也要会使用这些方法进行编程。

有关二级公共基础知识的考试要点以及基本概念,本书都作了简单介绍,一些基本术语在本书中都能找到。只要用好本书,相信考生一定能顺利通过二级公共基础部分的考试,也能帮助考生考好二级中的具体科目。

1.2 应试技巧

根据大纲要求,二级公共基础在二级各科笔试中占30%。其中选择题10个,每题2分,共20分;填空题5题,每题2分,共10分。

二级公共基础所包括的内容很多,在二级各科中所占的比例也不小。如何掌握好这些内容,特别是抓住考试的重点,在考试中获得好成绩,这是考生最关心的问题,也是编写本书的主要目的。

本书根据二级公共基础中的四部分内容分为四章。每一章均分为基本要点、典型例题分析、同步综合练习和同步综合练习参考答案等四个专题。这四个专题实际上是由浅入深地对考生进行针对性辅导。

在基本要求这个专题中,简明扼要地叙述了大纲中对这部分内容的基本要求,它实际上是大纲中对这部分内容要求的细化,也反映了考核的基本内容和深度。考生对这个专题中所叙述的内容必须要掌握。

在典型例题分析这个专题中,以例题的方式进一步阐述了本章所要掌握的基本概念以及本章中一些主要的知识点,也反映了实际考试的形式。在这个专题中,对每一个题目都进行了详细的讲解。通过对这个专题中的例题分析和讲解,考生应该对这部分考试要点又有进一步的理解,也要学会如何在考试中作出正确的判断。并且,在这些给出的例题中,还可能补充了一些与主要知识点有关的其他相关知识。

如果在典型例题分析这个专题中,考生是在被动的接受知识,而很少去主动的思考,那么在同步综合练习这个专题中,就要求考生要主动思考,完成其中给出的练习。如果你能独立完成这些练习,则说明基本掌握了考试大纲中对这部分内容的要求;如果对于某些题目还不太理解或没有把握,则可以参看最后一个专题中对这些练习的分析和答案。

如果从考试大纲可以了解考试中对二级公共基础的基本要求和考试内容,从相应的教材中可以学习到大纲中所要求的基本内容,那么,从本辅导书中可以进一步了解到考试内容的深度和广度以及考生如何去应试。这就是本书的宗旨。

1.1 基本要点

本章的考试要点如下：

- (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)。

1.1.1 算法的基本概念

1. 算法与数据结构的关系

程序设计主要包括两个方面：一是行为特性的设计；二是结构特性的设计。

行为特性的设计一般是指将解决问题过程中的每一个细节准确地加以定义，并将全部的解题过程用某种工具完整地描述出来。这一过程也称为算法的设计。

结构特性的设计是指为问题的解决确定合适的数据结构。

数据结构与算法之间有着密切的关系。特别是对于数据处理问题，算法的效率通常与数据结构在计算机中的表示有着直接的关系。

2. 算法的基本特征

所谓算法是指解题方案的准确而完整的描述。

通常，算法又分为数值型算法与非数值型算法。非数值型算法又称为符号处理。

对于一个问题，如果可以通过一个计算机程序，在有限的存储空间内运行有限长的时间而得到正确的结果，则称这个问题是算法可解的。但算法不等于程序，也不等于计算方法。当然，程序也可以作为算法的一种描述，但程序通常还需考虑很多与方法和分析无关的细节问题，这是因为在编写程序时要受到计算机系统运行环境的限制。

算法实际上是一种抽象的解题方法，它具有动态性。作为一个算法，一般应具有以下

几个基本特征:

(1) 能行性(effectiveness)

算法的能行性主要包括两个方面:一是算法中的每一个步骤必须是能实现的。例如在算法中,不允许出现分母为零的情况,在实数范围内不能求一个负数的平方根等。二是算法执行的结果要能达到预期的目的。通常,针对实际问题设计的算法,人们总是希望能够得到满意的结果。因此,在设计一个算法时,必须要考虑它的能行性,否则是不会得到满意结果的。

(2) 确定性(definiteness)

算法的确定性是指算法中的每一个步骤都必须是有明确定义的,不允许有模棱两可的解释,也不允许有多义性。这一性质也反映了算法与数学公式的明显差别。

(3) 有穷性(finiteness)

算法的有穷性是指算法必须能在有限的时间内做完,即算法必须能在执行有限个步骤之后终止。

算法的有穷性还应包括合理的执行时间的含义。因为,如果一个算法需要执行千万年,也失去了实用价值。

(4) 拥有足够的情报

一个算法是否有效,还取决于为算法所提供的情报是否足够。通常,算法中的各种运算总是要施加到各个运算对象上,而这些运算对象又可能具有某种初始状态,这是算法执行的起点或依据。因此,一个算法执行的结果总是与输入的初始数据有关,不同的输入将会有不同的结果输出。当输入不够或输入错误时,算法本身也就无法执行或导致执行有错。一般来说,当算法拥有足够的情报时,此算法才是有效的,而当提供的情报不够时,算法并不有效。

综上所述,所谓算法,是一组严谨地定义运算顺序的规则,并且每一个规则都是有效的且是明确的,此顺序将在有限的次数下终止。

3. 算法的基本要素

一个算法通常由两种基本要素组成:一是对数据对象的运算和操作,二是算法的控制结构。

(1) 算法中对数据的运算和操作

每个算法实际上是按解题要求从环境能进行的所有操作中选择合适的操作所组成的一组指令序列。因此,计算机算法就是计算机能处理的操作所组成的指令序列。

通常,计算机可以执行的基本操作是以指令的形式描述的。一个计算机系统能执行的所有指令的集合,称为该计算机系统的指令系统。计算机程序就是按解题要求从计算机指令系统中选择合适的指令所组成的指令序列。

(2) 算法的控制结构

一个算法的功能不仅取决于所选用的操作,而且还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。

算法的控制结构给出了算法的基本框架,它不仅决定了算法中各操作的执行顺序,而

且也直接反映了算法的设计是否符合结构化原则。描述算法的工具通常有传统流程图、N-S结构化流程图、算法描述语言等。一个算法一般都可以用顺序、选择、循环三种基本控制结构组合而成。

4. 算法设计基本方法

计算机解题的过程实际上是在实施某种算法,这种算法称为计算机算法。

常用的算法设计方法有:

(1) 列举法

列举法的基本思想是根据提出的问题,列举所有可能的情况,并用问题中给定的条件检验哪些是需要的,哪些是不需要的。因此,列举法常用于解决“是否存在”或“有多少种可能”等类型的问题,例如求解不定方程的问题。

列举法的特点是算法比较简单,但当列举的可能情况较多时,执行列举算法的工作量将会很大。因此,在用列举法设计算法时,使方案优化,尽量减少运算工作量,是应该重点注意的。通常,在设计列举算法时,只要对实际问题进行详细的分析,将与问题有关的知识条理化、完备化、系统化,从中找出规律;或对所有可能的情况进行分类,引出一些有用的信息,是可以大大减少列举量的。

列举原理是计算机应用领域中十分重要的原理。许多实际问题,若采用人工列举是不可想象的,但由于计算机的运算速度快,擅长重复操作,可以很方便地进行大量列举。列举算法虽然是一种比较笨拙而原始的方法,其运算量比较大,但在有些实际问题中(如寻找路径、查找、搜索等问题),局部的使用列举法却是很有有效的,因此,列举算法是计算机算法中的一个基础算法。

(2) 归纳法

归纳法的基本思想是通过列举少量的特殊情况,经过分析,最后找出一般的关系。显然,归纳法要比列举法更能反映问题的本质,并且可以解决列举量为无限的问题。但是,从一个实际问题中总结归纳出一般的关系,并不是一件容易的事情,尤其是要归纳出一个数学命题更为困难。从本质上讲,归纳就是通过观察一些简单而特殊的情况,最后总结出有用的结论或解决问题的有效途径。

(3) 递推

所谓递推,是指从已知的初始条件出发,逐次推出所要求的各中间结果和最后结果。其中初始条件或是问题本身已经给定,或是通过对问题的分析与化简而得到确定。递推本质上属于归纳法,工程上许多递推关系式实际上是通过在实际问题的分析与归纳而得到的,因此,递推关系式往往是归纳的结果。

(4) 递归

递归分为直接递归与间接递归两种。如果一个算法 P 显式地调用自己则称为直接递归。如果算法 P 调用另一个算法 Q,而算法 Q 又调用算法 P,则称为间接递归调用。

递归是一种很重要的算法设计方法之一。实际上,递归过程能将一个复杂的问题归结为若干个较简单的问题,然后将这些较简单的每一个问题再归结为更简单的问题,这个过程可以一直做下去,直到最简单的问题为止。

有些实际问题,既可以归纳为递推算法,又可以归纳为递归算法。但递推与递归的实现方法是大不一样的。递推是从初始条件出发,逐次推出所需求的结果;而递归则是从算法本身到达递归边界的。通常,递归算法要比递推算法清晰易读,其结构比较简练。特别是在许多比较复杂的问题中,很难找到从初始条件推出所需结果的全过程,此时,设计递归算法要比递推算法容易得多。但递归算法的执行效率比较低。

(5) 减半递推技术

解决实际问题的复杂程度往往与问题的规模有着密切的关系。因此,利用分治法解决这类实际问题是有效的。所谓分治法,就是对问题分而治之。工程上常用的分治法是减半递推技术。

所谓“减半”,是指将问题的规模减半,而问题的性质不变。所谓“递推”。是指重复“减半”的过程。

(6) 回溯法

在工程上,有些实际问题很难归纳出一组简单的递推公式或直观的求解步骤,并且也不能进行无限的列举。对于这类问题,一种有效的方法是“试”。通过对问题的分析,找出一个解决问题的线索,然后沿着这个线索逐步试探,对于每一步的试探,若试探成功,就得到问题的解,若试探失败,就逐步回退,换别的路线再进行试探。这种方法称为回溯法。回溯法在处理复杂数据结构方面有着广泛的应用。

5. 算法复杂度

算法的复杂度主要包括时间复杂度和空间复杂度。

(1) 算法的时间复杂度

所谓算法的时间复杂度,是指执行算法所需要的计算工作量。

为了能够比较客观地反映出一个算法的效率,在度量一个算法的工作量时,不仅应该与所使用的计算机、程序设计语言以及程序编制者无关,而且还应该与算法实现过程中的许多细节无关。为此,可以用算法在执行过程中所需基本运算的执行次数来度量算法的工作量。基本运算反映了算法运算的主要特征,因此,用基本运算的次数来度量算法工作量是客观的也是实际可行的,有利于比较同一问题的几种算法的优劣。

算法的工作量用算法所执行的基本运算次数来度量,而算法所执行的基本运算次数是问题规模的函数,即

$$\text{算法的工作量} = f(n)$$

其中 n 是问题的规模。例如,两个 n 阶矩阵相乘所需要的基本运算(即两个实数的乘法)次数为 n^3 ,即计算工作量为 n^3 ,也就是时间复杂度为 n^3 。

在同一问题规模下,如果算法执行所需的基本运算次数取决于某一特定输入时,可以用以下两种方法来分析算法的工作量。

① 平均性态(Average Behavior)

所谓平均性态分析,是指用各种特定输入下的基本运算次数的带权平均值来度量算法的工作量。

设 x 是所有可能输入中的某个特定输入, $p(x)$ 是 x 出现的概率(即输入为 x 的概

率), $t(x)$ 是算法在输入为 x 时所执行的基本运算次数, 则算法的平均性态定义为

$$A(Nn) = \sum_{x \in D_n} p(x) t(x)$$

其中 D_n 表示当规模为 n 时, 算法执行时所有可能输入的集合。这个式子中的 $t(x)$ 可以通过分析算法来加以确定; 而 $p(x)$ 必须由经验或用算法中有关的一些特定信息来加以确定, 通常是不能解析地加以计算的。

②最坏情况复杂性(Worst-Case Complexity)

所谓最坏情况分析, 是指在规模为 n 时, 算法所执行的基本运算的最大次数。它定义为

$$W(n) = \max_{x \in D_n} \{t(x)\}$$

显然, $W(n)$ 的计算要比 $A(n)$ 的计算方便得多。由于 $W(n)$ 实际上是给出了算法工作量的一个上界, 因此, 它比 $A(n)$ 更具有实用价值。

(2) 算法的空间复杂度

一个算法的空间复杂度, 一般是指执行这个算法所需要的内存空间。

一个算法所占用的存储空间包括算法程序所占的空间、输入的初始数据所占的存储空间以及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(例如在链式结构中, 除了要存储数据本身外, 还需要存储链接信息)。如果额外空间量相对于问题规模来说是常数, 则称该算法是原地(in place)工作的。在许多实际问题中, 为了减少算法所占的存储空间, 通常采用压缩存储技术, 以便尽量减少不必要的额外空间。

1.1.2 数据结构的基本概念

数据结构作为计算机的一门学科, 主要研究和讨论以下三个方面的问题:

- (1) 数据集合中各数据元素之间所固有的逻辑关系, 即数据的逻辑结构;
- (2) 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;
- (3) 对各种数据结构进行的运算。

1. 什么是数据结构

简单地说, 数据结构是指相互有关联的数据元素的集合。

在数据处理领域中, 每一个需要处理的对象都可以抽象成数据元素。数据元素一般简称为元素。

在具有相同特征的数据元素集合中, 各个数据元素之间存在有某种关系(即联系), 这种关系反映了该集合中的数据元素所固有的一种结构。在数据处理领域中, 通常把数据元素之间这种固有的关系简单地用前后件关系(或直接前驱与直接后继关系)来描述。

前后件关系是数据元素之间的一个基本关系, 但前后件关系所表示的实际意义是随具体对象的不同而不同。一般来说, 数据元素之间的任何关系都可以用前后件关系来描述。

(1) 数据的逻辑结构

数据结构是指反映数据元素之间关系的数据元素集合的表示。更通俗地说, 数据结

构是指带有结构的数据元素的集合。在此,所谓结构实际上就是指数据元素之间的前后件关系。

一个数据结构应包含以下两方面的信息:

- (1)表示数据元素的信息;
- (2)表示各数据元素之间的前后件关系。

在以上所述的数据结构中,其中数据元素之间的前后件关系是指它们的逻辑关系,而与它们在计算机中的存储位置无关。因此,上面所述的数据结构实际上是数据的逻辑结构。

所谓数据的逻辑结构,是指反映数据元素之间逻辑关系的数据结构。

由前面的叙述可以知道,数据的逻辑结构有两个要素:一是数据元素的集合,通常记为 D ;二是 D 上的关系,它反映了 D 中各数据元素之间的前后件关系,通常记为 R ,即一个数据结构可以表示成

$$B=(D,R)$$

式中 B 表示数据结构。为了反映 D 中各数据元素之间的前后件关系,一般用二元组来表示。例如假设 a 与 b 是 D 中的两个数据,则二元组 (a,b) 表示 a 是 b 的前件, b 是 a 的后件。这样,在 D 中的每两个元素之间的关系都可以用这种二元组来表示。

(2) 数据的存储结构

数据的逻辑结构在计算机存储空间中的存放形式称为数据的存储结构(也称数据的物理结构)。

一个数据结构中的各数据元素在计算机存储空间中的位置关系与逻辑关系有可能是不同的。由于数据元素在计算机存储空间中的位置关系可能与逻辑关系不同,因此,为了表示存放在计算机存储空间中的各数据元素之间的逻辑关系(即前后件关系),在数据的存储结构时,不仅要存放各数据元素的信息,还需要存放各数据元素之间的前后件关系的信息。

一般来说,一种数据的逻辑结构根据需要可以表示成多种存储结构,常用的存储结构有顺序、链接、索引等存储结构。而采用不同的存储结构,其数据处理的效率是不同的。因此,在进行数据处理时,选择合适的存储结构是很重要的。

2. 数据结构的图形表示

在数据结构的图形表示中,对于数据集合 D 中的每一个数据元素用中间标有元素值的方框表示,一般称之为数据结点,并简称为结点;为了进一步表示各数据元素之间的前后件关系,对于关系 R 中的每一个二元组,用一条有向线段从前件结点指向后件结点。

例如,一年四季的数据结构可以用如图 1.1 所示的图形来表示。

又如,反映家庭成员间辈分关系的数据结构可以用如图 1.2 所示的图形表示。

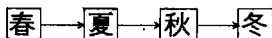


图 1.1 一年四季数据结构的图形表示