

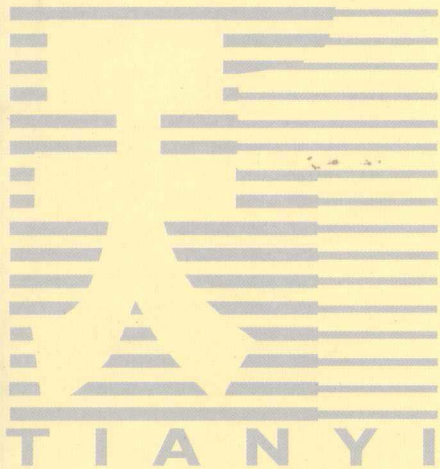
计算机类


依据国家自考委最新自考大纲及新修版教材编写

数据结构导论

● 高等教育自学考试指定教材同步配套题解

主编 丁峻岭 王洛民



 现代出版社

2014年1月1日

致 謝 詞

2014年1月1日

2014年1月1日

高等教育自学考试指定教材同步配套题解

(计算机及其应用专业)

数据结构导论

主 编 丁峻岭 王洛民
副主编 陈有才 孙 建
李学相

现代出版社

图书在版编目(CIP)数据

数据结构导论/丁峻岭 王洛民 主编—北京:现代出版社,2000.12
全国高等教育自学考试指定教材同步配套题解,计算机专业辅导

ISBN 7-80028-603-7

I.高… II.①丁…②王… III.数据结构导论-高等教育-自学考试-解题
IV.G726.9

中国版本图书馆 CIP 数据核字(2000)第 85296 号

高等教育自学考试指定教材同步配套题解

数据结构导论

责任编辑:姜秀云

出版发行:现代出版社

地 址:北京安定门外安华里 504 号(100011)

印 刷:中国科学院印刷厂

开 本:787×1092 1/16

版 次:2000 年 12 月第 1 版 2002 年 4 月第 2 次印刷

印 数:10001-20000 册

印 张:15.5 印张

书 号:ISBN 7-80028-603-7/G.205

定 价:268.00(16 册)

本册定价:22.00 元

(本书封面贴有防伪标签,无标签者均为盗版)

寄语考生

随着我国教育总方针由应试教育向素质教育的转变,作为我国高等教育重要组成部分的自学考试也发生了重大变化。全国自考委在专业设置、考试计划、出题指导方针等方面都做了重大的调整,同时,对自学考试大纲、指定教材亦做了全方面的修订、编写。

新形势下,为使广大自考学员能及时、快速地掌握新教材,我们对原有的系列辅导用书进行了全面的修订,并不断地推出新品种以飨读者。

本套“指定教材同步配套题解”有以下特点:

新—①内容新。本套丛书全部按最新的自学考试大纲及最新版指定教材内容编写。

②结构新。同原辅导及其它辅导相比,修订后的辅导用书编排体例更加科学,增加了“本门课的学习与考试”部分。这是全书的点睛之笔。

全—信息全。本套辅导书涵盖了大纲中所有的知识点、考核点,并精心编拟大量“综合练习题”,训练强度大,解答准确。特别指出的是根据《高等教育自学考试活页文丛》(人大版)对教材中没有补充的内容,在本辅导中都做了详尽补充。

强—①作者阵容强。本套丛书的作者,有指定教材的主编,有专业教研室主任,有长期参加辅导的主讲教师。他们对自考教材分析透,对出题规律掌握准。

②针对性强。书后针对新大纲及考卷合理设计多套“全真模拟试题”,增强考生临场经验,增加本书实用性。

愿本套“同步配套题解”能帮助您顺利通过自考难关,早日实现美好理想。

《高等教育自学考试指定教材同步配套题解》编委会

前 言

《数据结构导论同步配套题解》一书,以全国高等教育自学考试指导委员会指定的《数据结构导论》(计算机及应用专业·专科、全国高等教育自学考试指导委员会组编、陈小平主编、经济科学出版社 2000 年版)课本为基础,仅仅围绕全国高等教育自学考试指导委员会制定的计算机及应用专业(专科)《数据结构导论自学考试大纲》的要求编撰而成。

全书主要有序言部分“《数据结构导论》课程的学习与考试”、第一至八章的学习重点及同步综合练习和尾部所附的全真模拟试题三部分组成。中间部分是全书的核心,具体到各章又分“学习目的与要求”、“考核知识点”、“考核内容”、“同步综合练习”、“参考答案”五部分。“同步综合练习”从“名词解释”、“填空”、“单项选择”、“简答及应用”、“算法设计”等五个方面提出了问题,参考答案回答了这五个方面的提问。

用数字计算机解决任何实际问题都离不开数据表示和数据处理,而数据的表示和处理的核心问题之一是数据结构及其实现——这正是数据结构课程的基本内容。从这个意义上说,《数据结构导论》课程在知识学习和技能培养两个方面都处于关键性地位。

特别注意的是,《数据结构导论》课程的技能性很强。系统地学习和掌握在不同存储结构上实现的不同算法及其设计思想,从中体会并掌握结构选择和算法设计的思维方式及技巧,提高分析问题和解决问题的能力,是应考者须具备的最基本要求。书上的习题不少通过改写是可以上机练习的,应注意结合本课程学习进度,进行上机实验,通过编写程序,巩固加深对各知识点的学习和理解。

本着对读者认真负责的态度,该书在编写过程中严格依据“大纲”要求,突出指定“教材”的内容,加强基础知识的学习与指导,适当增加了一些“大纲”中提到而指定“课本”中未涉及或涉及较浅的一些知识点。特别注重引导考生把基础知识和理论转化为应用能力,尽最大可能指导、帮助考生顺利通过对《数据结构导论》课程的考试。

由于时间仓促,加之水平有限,书中错误与不足之处在所难免,恳请广大读者批评指正,以利日后改进。

编 者

目 录

《数据结构导论》课程的学习与考试	(1)
第一章 概论	(5)
学习目的与要求	(5)
考核内容	(6)
同步综合练习	(10)
参考答案	(15)
第二章 线性表	(19)
学习目的与要求	(19)
考核内容	(20)
同步综合练习	(30)
参考答案	(42)
第三章 栈、队列	(57)
学习目的与要求	(57)
考核内容	(58)
同步综合练习	(66)
参考答案	(78)
第四章 树	(91)
学习目的与要求	(91)
考核内容	(92)
同步综合练习	(100)
参考答案	(112)
第五章 图	(134)
学习目的与要求	(134)
考核内容	(134)
同步综合练习	(140)
参考答案	(147)
第六章 查找表	(158)
学习目的与要求	(158)
考核内容	(159)
同步综合练习	(166)
参考答案	(174)
第七章 文件	(186)
学习目的与要求	(186)
考核内容	(186)

同步综合练习	(188)
参考答案	(193)
第八章 排序	(197)
学习目的与要求	(197)
考核内容	(197)
同步综合练习	(201)
参考答案	(206)
全真模拟试题(一)	(218)
参考答案	(222)
全真模拟试题(二)	(226)
参考答案	(229)
2001年下半年全国高教自考数据结构导论试题	(234)
参考答案	(238)

《数据结构导论》课程的学习与考试

一、课程性质及其设置目的和要求

(一)课程的性质、地位和任务

《数据结构导论》课程是高等教育自学考试计算机及应用专业(专科)的一门重要的专业基础课。用数字计算机解决任何实际问题都离不开数据表示和数据处理,而数据的表示和处理的核心问题之一是数据结构及其实现——这正是数据结构课程的基本内容。从这个意义上说,数据结构课程在知识学习和技能培养两个方面都处于关键性地位。本课程不仅为数据库及其应用、操作系统概论等后继软件课程提供了必要的知识基础,也为计算机及应用的专业人员提供了必要的技能训练。

(二)本课程的要求

通过本课程的学习,要求应考者应达到知识和技能两方面的目标:

1. 知识方面:从数据结构及其实现这两个层次及其相互关系的角度,系统地学习和掌握常用基本数据结构(包括线性表、栈、队列、二叉树、图、查找表和文件等)及其不同的实现(包括不同的存储结构和算法),了解并掌握分析、比较和选择不同数据结构及不同存储结构、不同运算实现(即算法)的原则和方法,为后继课程的学习打好基础。

2. 技能方面:系统地学习和掌握在不同存储结构上实现的不同算法及其设计思想,从中体会并掌握结构选择和算法设计的思维方式及技巧,提高分析问题和解决问题(包括编程等)的能力。

(三)本课程与有关课程的联系

本课程的先修课程包括高级语言程序设计和汇编语言程序设计,后继课程有数据库及其应用和操作系统概论。它们之间的关系是:

1. 本课程中存储结构或存储结点的类型定义、实现运算的算法特别是指针操作,需要以高级语言(C语言)的类型和变量说明、指针型变量及其操作,以及程序设计(和阅读)等基础知识和基本技能为前提。另一方面,本课程详细介绍的各种基本概念、基本数据结构及其实现,为数据库及其应用和操作系统概论提供了重要的基础和工具。

2. 高级语言程序设计课程只在高级语言层次上涉及数据的表示和处理。本课程首次在抽象层次并以抽象层次与语言层次相联系的方式,系统地讨论了数据表示和数据处理的若干基本问题。而后继软件课程则以本课程为基础,在更高的层次和更大的规模上研究数据的表示和处理。因此,本课程是数据表示和处理这条主线上的一个必不可少的环节。

二、课程的学习与考核

(一)能力层次的含义

在本书第一至八章“考核知识点”的考核要求中,提出了“识记”、“领会”、“简单应用”、“综合应用”等四个能力层次。它们之间是递进等级的关系。后者必须建立在前者基础

上,它们的含义是:

“识记”:要求考生能够识别和记忆本课程中规定的有关知识点的主要内容(如定义、定理、定律、表达式、公式、原则、重要结论、方法、步骤及特征、特点等),并能够根据考核的不同要求,做出正确的表述、选择和判断。

“领会”:要求考生能够领悟和理解本课程中规定的有关知识点的内涵与外延,熟悉其内容要点和它们之间的区别与联系,并能够根据考核的不同要求,做出正确的解释、说明和论述。

“简单应用”:要求考生能够运用本课程中规定的少量知识点,分析和解决一般应用问题,如简单的计算、绘图和分析、论证等。

“综合应用”:要求考生能够运用本课程中规定的多个知识点,分析和解决较复杂的应用问题,如计算、绘图、简单设计、编程和分析、论证等。

(二)自学教材

自学教材:《数据结构导论》,全国高等教育自学考试指导委员会组编,陈小平主编,经济科学出版社 2000 年版。

(三)课程学分

本课程共 4 学分。

(四)关于命题和考试的一些规定

1. 考试采用闭卷笔试方式,时间 150 分钟。考试时无需使用笔和橡皮之外的任何器具。

2. 本书各章“考核知识点”中所列各知识点内的细目均属考试内容。试题覆盖到章,适当突出重点章节,加大重点内容的覆盖密度。

本课程的命题考试,应根据本课程大纲(本书第一至八章的“考核知识点”体现了该大纲的要求)所规定的考试内容和考试目标(考核知识点、考核要求)来确定考试范围和考核要求,不要任意扩大或缩小考试范围,提高或降低考试要求。考试命题要覆盖到各章,并适当突出重点章节,体现本课程的内容重点。

3. 试卷中对不同能力层次要求的试题所占比例大致为:识记占 20%,领会占 30%,简单应用 30%,综合应用 20%。

4. 试题的难易程度与能力层次不是一个概念,它们之间有一定的关联,但并不是完全吻合。在各个能力层次中对于不同的考生都存在着不同的难度。要合理地安排试题的难易程度,试题难度可分为:易、较易、较难和难四个等级,每份试卷中不同难度试题的分数所占的比例一般为 2:3:3:2。

5. 试题的题型可以有:单项选择题、填空题、应用题、算法设计等。

三、自学指导

初学者往往感到数据结构课程的内容多、难度大。努力做到以下几点有助于改善自学效果。

1. 注意知识体系。数据结构课程中的知识本身具有良好的结构性。从总体上说,课程的主要内容是围绕着线性表、串、栈、队列、数组、二叉树、图、查找表和文件这九种常用的数据结构和排序这种常用运算来组织的,对每种数据结构又是从“定义”(包括逻辑结构和基本运算二个部分)和“实现”(包括存储结构和运算实现两个方面)这两个层次以及它

们之间的联系的角度加以介绍的。对排序运算,讨论了它的各种典型、常见的实现算法。按上述体系对课程中的具体内容加以分类,有助于整体上的全面把握。

2.注意比较。正由于本课程中的知识点具有如上所述的体系,自学中应该从“纵向”和“横向”两个方面对比有关内容以便加深理解。纵向对比包括将一种数据结构与它的各种不同的实现加以比较;横向对比包括具有相同逻辑结构的不同数据结构(如线性表、栈、队列和串)的比较,同一数据结构的存储结构和实现同一运算的不同算法(如各种查找算法)的比较,等等。

3.注意复习和重读。有些内容在初读时难以透彻理解或熟练掌握。在继续学习的过程中遇到有关内容时,应及时重读或复习,这往往能够化难为易、温故知新。

4.充分利用自考大纲。在进入每章之前和结束每章之后,仔细阅读大纲的有关规定和要求,有利于集中思路和自我检查。

5.注意循序渐进。在进入具体内容(如存储结构或算法)之前,首先领会基本概念、基本思想是极为重要的。特别是在阅读算法之前,一定要先弄清其基本思想、基本步骤,这将大大降低理解算法的难度;而且,读“懂”其算法而不知其基本思想,远不能算是真懂。

6.注意练习。习题是本课程的重要组成部分,只看书不做题是不可能真正学会有关知识,更不能达到技能培养的目标的。同时,做习题也是自我检查的重要手段。此外,在做算法设计型习题时不要直接调用书上已写有函数的算法(标准函数除外),而应独立设计出完整的算法,以利于编程能力的提高。

四、本书的内容设置及使用

本书以全国高等教育自学考试指导委员会指定的《数据结构导论》(计算机及应用专业·专科、全国高等教育自学考试指导委员会组编、陈小平主编、经济科学出版社2000年版)课本为基础,仅仅围绕全国高等教育自学考试指导委员会制定的计算机及应用专业(专科段)《数据结构导论自学考试大纲》的要求编撰而成。

全书主要有序言部分《数据结构导论》课程的学习与考试、第一至八章的学习重点及综合练习和尾部所附的全真模拟试题三部分组成。中间部分是全书的核心,具体到各章又分“学习目的与要求”、“知识网点”、“综合练习”、“参考答案”四部分。知识网点含考核知识点和各节的重点内容提要。综合练习从“名词解释”、“填空”、“单项选择”、“简答及应用”、“算法设计”等五个方面提出了问题,参考答案回答了这五个方面的提问。

编写过程中,“序言”部分及第一至八章的“学习目的与要求”、“考核知识点”,主要借鉴了《数据结构导论自学考试大纲》相应部分的一些要求及说明。对于“自考委”指定课本《数据结构导论》一书所附习题,作为重点内容在本书相应章节的综合练习及参考答案中进行了体现。

为使各考生更好地学习《数据结构导论》课程,引导考生把基础知识和理论转化为应用能力,帮助考生顺利通过《数据结构导论》课程的考试,本书在编写过程中重点突出了以下几点:

1.严格依据“大纲”要求。编写过程中,“序言”部分及第一至八章的“学习目的与要求”、“考核知识点”,主要借鉴了《数据结构导论自学考试大纲》相应部分的一些要求及说明。

2.突出指定的“教材”内容。对于“自考委”指定课本《数据结构导论》一书所附习题,

作为重点内容在本书相应章节的综合练习及参考答案中进行了体现。

3.加强基础知识练习。对于“大纲”及课本中涉及的一些重点内容、基础知识,在考核内容及综合练习中依多种形式进行了练习。

4.适当增加知识面。对于“大纲”中提到而指定“课本”中未涉及或涉及不深的一些知识点,在本辅导书的编写中,于考核内容及综合练习中进行了适当的补充。

5.确保答题质量。

第一章 概 论

学习目的与要求

本章集中介绍贯穿和应用于数据结构课程始终的基本概念和主要工具,概括反映了后继各章的基本问题,为进入具体内容的学习提供了必要的引导。

本章总的要求是:理解数据、数据元素和数据项的概念及其相互关系;理解逻辑结构、基本运算和数据结构的概念、意义和分类;理解存储结构与逻辑结构的关系;了解机内表示的级别和四种基本存储方式;理解算法的概念;了解算法分析的基本概念、时间复杂性及其量级的概念。

本章重点是逻辑结构和数据结构的概念。难点是算法的时间复杂性分析。

考核知识点

1. 数据、数据元素和数据项的概念,要求达到“领会”层次。
 - 1.1 数据的含义。
 - 1.2 数据元素和数据项的概念。
2. 数据的逻辑结构,要求达到“领会”层次。
 - 2.1 逻辑结构的意义。
 - 2.2 数据元素间逻辑关系的含义。
 - 2.3 四类基本逻辑结构(集合、线性结构、树形结构和图状结构)的不同特点。
3. 运算的概念,要求达到“识记”层次。
 - 3.1 运算是在逻辑结构层次上对处理功能的抽象。
 - 3.2 基本运算的含义和作用。
4. 数据结构的概念,要求达到“领会”层次。
 - 4.1 数据结构的含义。
 - 4.2 数据结构涉及数据表示和数据处理两个方面。
5. 存储结构和运算实现,要求达到“识记”层次。
 - 5.1 存储结构的含义和四种基本存储方式的基本思想。
 - 5.2 运算实现与运算的联系与区别。
6. 算法及其描述,要求达到“简单应用”层次。
 - 6.1 算法的概念和分类,能用类 C 语言描述算法。
7. 算法分析,要求达到“识记”层次。
 - 7.1 算法在给定输入下的计算量的含义和估算的方法。
 - 7.2 最坏情况时间复杂性和平均时间复杂性的概念。
 - 7.3 空间复杂性的含义。

考核内容

第一节 引言

程序设计的实质是数据表示和数据处理,而这种表示和处理应通过一个渐进的过程逐步完成。

用数字式计算机解决问题的实质是对数据的加工处理。数据在计算机存储器中的存在形式称为机内表示。为了让计算机去加工处理数据,必须首先将数据从机外表示转化为机内表示。这项任务称为数据表示。用适当的可执行语句编制程序,以便让计算机去执行对数据机内表示的各种操作,从而实现处理要求,即得到所需的结果,这项工作称为数据处理。计算机专业人员必须完成的两项基本任务是:数据表示和数据处理。

概括地说,数据结构课程的主要内容包括:数据的逻辑结构、定义在逻辑结构上的基本运算、数据的存储结构和运算的实现。其中,数据的逻辑结构是数据的组织形式,基本运算规定了数据的基本操作方式。由一种逻辑结构和一组基本运算构成的整体是实际问题的一种数学模型,这种数学模型的建立、选择和实现是数据结构的核心问题。存储结构是逻辑结构的存储实现,即数据按逻辑结构规定的形式在计算机存储器中的存放方式。运算实现是完成运算功能的算法,或这些算法的设计。此外,本课程还要考虑各种结构和实现方法的评价和选择。

程序设计是一个渐进的过程,需注意以下三点:

- (1)数据表示任务是逐步完成的,即数据表示形式的变化过程是:机外表示→逻辑结构→存储结构。
- (2)数据处理任务也是逐步完成的,即有转化过程:处理要求→基本运算和运算→算法。
- (3)数据表示与数据处理是密切相关的,数据处理方式总是与数据的某种相应的表示形式相联系,反之亦然。

第二节 数据、逻辑结构和运算

从数据结构的观点看,通常所说的“数据”应分成三个不同的层次,即数据、数据元素和数据项。

一、数据、数据元素和数据项

凡能被计算机存储、加工的对象通称为数据。

数据元素是数据的基本单位,在程序中作为一个整体而加以考虑和处理。换句话说,数据元素被当作运算的基本单位,并且通常具有完整确定的实际意义。根据需要,数据元素又被称为元素、结点、顶点或记录。

在很多情况下,数据元素又是由数据项组成的,但数据项通常不具有完整确定的实际意义,或不被当作一个整体对待。在有些场合下,数据项又称为字段或域。它是数据的不可分割的最小标识单位。

从某种意义上说,数据、数据元素和数据项实际反映了数据组织的三个层次,数据可由若干个数据元素构成,而数据元素又可由若干个数据项构成。

二、数据的逻辑结构

所谓逻辑关系是指数据元素之间的关联方式或称“邻接关系”。数据元素之间逻辑关系的整体称为逻辑结构。数据的逻辑结构就是数据的组织形式。

在任何问题中,数据元素都不是孤立存在的,它们之间总是存在着某种关系(集合可以看成是没有关系的关系),称其为结构。根据数据元素之间关系的不同特性,通常有集合、线性结构、树形结构、图状结构四类基本逻辑结构,它们反映了四类基本的数据组织形式。集合中任何两个结点之间都没有逻辑关系,组织形式松散。线性结构中结点按逻辑关系依次排列形成一条“锁链”。树形结构具有分支、层次特性,其形态有点像自然界中的树。图状结构最复杂,其中的各个结点按逻辑关系互相缠绕,任何两个结点都可以邻接。

关于逻辑结构,有以下几点需特别注意:

- (1)逻辑结构与数据元素本身的形式、内容无关。
- (2)逻辑结构与数据元素的相对位置无关。
- (3)逻辑结构与所含结点个数无关。

由此可见,一些表面上很不相同的数据可以有相同的逻辑结构,因此,逻辑结构是数据组织的某种“本质性”的东西。

三、运算和基本运算

一般地,运算是指在任何逻辑结构上施加的操作,即对逻辑结构的加工。这种加工以一个或多个逻辑结构及其他有关参数为对象,以经过修改的逻辑结构或从原逻辑结构中提取的有关信息为结果。根据操作的效果,可将运算分成以下两种基本类型:

- ①加工型运算,其操作改变了原逻辑结构的“值”,如结点个数、某些结点的内容等;
- ②引用型运算,其操作不改变原逻辑结构,只从中提取某些信息作为运算的结果。

将以某种逻辑结构 S 为操作对象的运算称为“定义在 S 上的运算”,简称“ S 上运算”。一般地,可能存在同一逻辑结构 S 上的两个运算 A 和 B , A 的实现需要或可以利用 B ,而 B 的实现不需要利用 A 。在这种情况下,称 A 可以“归约”为 B 。假如 Γ 是 S 上的一些运算的集合, Δ 是 Γ 的一个子集,使得 Γ 中每一运算都可以“归约”为 Δ 中一个或多个运算,而 Δ 中任一运算不可归约为别的运算,则称 Δ 中运算为(相对于 Γ 的)基本运算。

相对而言,基本运算比非基本运算简单。一旦基本运算得到实现,非基本运算的实现就十分容易。

第三节 存储实现和运算实现

一、存储实现

存储实现的基本目标是建立数据的机内表示。由于已经建立的逻辑结构是设计人员根据解题需要选定的数据组织形式,因此存储实现建立的机内表示应遵循选定的逻辑结构。另一方面,由于逻辑结构不包括结点内容即数据元素本身的表示,因此存储实现的另一主要内容是建立数据元素的机内表示。按上述思路所建立的数据的机内表示称为数据的存储结构。一般地,一个存储结构包括以下三个主要部分:

- ①存储结点(在不致混淆时简称为结点),每个存储结点存放一个数据元素;
- ②数据元素之间关联方式的表示,也就是逻辑结构的机内表示;

③附加设施,如为便于运算实现而设置的“哑结点”等等。

存储结构的主要部分是数据元素之间关联方式的表示。通常,存储结点之间可以有四种关联方式,称为四种基本存储方式:

(1)顺序存储方式。每个存储结点只含一个数据元素。所有存储结点相继存放在一个连续的存储区里。用存储结点间的位置关系表示数据元素之间的逻辑关系。按这种方式表示逻辑关系的存储结构称为顺序存储结构。

(2)链式存储方式。每个存储结点不仅含有一个数据元素,还包含一组指针。每个指针指向一个与本结点有逻辑关系的结点,即用附加的指针表示逻辑关系。按这种方式组织起来的存储结构称为链式存储结构。

(3)索引存储方式。每个存储结点只含有一个数据元素,所有存储结点连续存放。此外增设一个索引表,索引表中的索引指示各存储结点的存储位置或位置区间端点。按这种方式组织起来的存储结构称为索引存储结构。

(4)散列存储方式。每个结点含有一个数据元素,各个结点均匀分布在存储区里,用散列函数指示各结点的存储位置或位置区间端点。相应的存储结构称为散列存储结构。

可用任何一种存储方式所规定的存储结点之间的关联方式来间接表达给定逻辑结构 S 中数据元素之间的逻辑关系。由此得到的存储结构,称为给定逻辑结构 S 的存储实现或存储映象。

二、运算实现

一个运算的实现是指一个完成该运算功能的程序。运算实现的核心是处理步骤的规定,即算法设计。一般地,一个算法规定了求解给定类型问题所需的所有“处理步骤”及其执行顺序,使得给定类型的任何问题能在有限时间内被机械地求解。

任何算法都必须用某种语言加以描述。根据描述算法的语言的不同,可将算法分为以下三类:

①运行终止的程序可执行部分。是用程序设计语言描述的算法,这种算法可直接在计算机上运行,从而使给定问题在有限时间内被机械地求解。这种算法有时可称为程序。

②伪语言算法。采用某种“伪程序设计语言”描述的算法称为伪语言算法。它不可直接在机器上运行(可在某种“抽象机”上运行),但容易编写和阅读。

③非形式算法。用自然语言(如汉语),同时可能还使用了程序设计语言或伪程序设计语言描述的算法称为非形式算法。

类 C 语言基本上是标准 C 语言的简化。类 C 语言与标准 C 语言的主要区别如下:

(1)局部量的说明可以省略(但形参表中及函数类型的说明需保留),重要的变量需在注解中用文字说明其类型和作用。

(2)分情形语句可以采用下述形式:

```
switch
{ case 条件 1:      语句序列 1;break;
  case 条件 2:      语句序列 2;break;
  .....
  case 条件 n:      语句序列 n;break;
```


default: 语句序列 n + 1;

其中“default: 语句序列 n + 1;”可以省略。

(3) 不含 goto 语句, 增加了一个出错处理语句 error(字符串), 其功能是终止它所在算法的执行并回送表示出错信息的字符串。

(4) 输入输出语句有:

输入语句 scanf([格式串], 变量 1, ..., 变量 n);

输出语句 printf([格式串], 变量 1, ..., 变量 n);

通常省略格式串。

(5) 类 C 语言的形参书写比标准 C 语言简单, 如 int abc(int a, int b, int, c) 可以简写为 int abc(int a, b, c)。

第四节 算法分析

通常从以下几个方面评价算法(包括程序)的质量:

① 正确性 算法应能正确地实现预定的功能(即处理要求)。

② 易读性 算法应易于阅读和理解, 以便于调试、修改和扩充。

③ 健壮性 当环境发生变化(如遇到非法输入)时, 算法能适当地做出反应或进行处理, 不会产生不需要的运行结果。

④ 高效性 即达到所需要的时空性能。

一个算法的时空性能是指该算法的时间性能(或时间效率)和空间性能(或空间效率), 前者是算法包含的计算量, 后者是算法需要的存储量。

通常采用下述办法来估算求解某类问题的各个算法在给定输入下的计算量:

① 根据该类问题的特点合理地选择一种或几种操作作为“标准操作”;

② 确定每个算法在给定输入下共执行了多少次标准操作, 并将此次数规定为该算法在给定输入下的计算量。

通常, 一个算法在不同输入下的计算量是不同的。则可用以下两种方式来确定一个算法的计算量:

① 以算法在所有输入下的计算量的最大值作为算法的计算量, 这种计算量称为算法的最坏情况时间复杂性或最坏情况时间复杂度。

② 以算法在所有输入下的计算量的加权平均值作为算法的计算量, 这种计算量称为算法的平均时间复杂性或平均时间复杂度。

最坏情况时间复杂性和平均时间复杂性统称为时间复杂性(或时间复杂度), 用 $T(n) = O(f(n))$ 表示。其中, $f(n)$ 是算法中频度最大的那条语句频度的数量级。

例如, 计算下列程序段的时间复杂度:

```
... for(i = 1; i <= n; i++) { k++; for(j = 1; j <= n; j++) l += k; } ...
```

对上述程序段, 语句执行的总次数为: $n + 1 + n + n(n + 1) + n^2 = 2n^2 + 3n + 1$ 。求解时间复杂度的方法为: 找出所有语句中执行频度最大的那条语句的频度, 然后取其数量级放入 O 中即可。得 $T(n) = O(f(n)) = O(n^2)$ 。