

全国高等教育自学考试应试指导丛书  
中国计算机函授学院图书编写中心 组编

计算机及应用专业  
(专科)

# 数据结构导论 自考应试指导

主 编 邹华跃  
副主编 王淮亭  
主 审 胡学钢



南京大学出版社

中国计算机函授学院图书编写中心 组编

全国高等教育自学考试应试指导丛书

计算机及应用专业(专科)

# 数据结构导论自考应试指导

主 编 邹华跃

副主编 王淮亭

主 审 胡学钢

南 京 大 学 出 版 社

## 内 容 简 介

本书是根据全国高等教育自学考试委员会公布的《数据结构导论自学考试大纲》的要求,结合作者多年的教学经验积累,精心编写而成的。全书共分八章,每章又分为三个部分:知识点、典型题解与分析、课后习题解答与分析。其中,知识点部分涵盖大纲,概念清楚,讲解详细,通俗易懂;典型题解与分析部分严格按考试大纲的要求精心挑选了150余题,既具有一定的代表性,又避免陷入题海而过多增加考生的负担;课后习题分析与解答部分对指定教材《数据结构导论》上的习题进行了全面、细致的分析与解答。最后提供了三套模拟试卷,并进行了分析与解答。

本书结构清晰,针对性强,覆盖面广,不仅对考试内容进行了高度概括,还通过分析、解答的形式提供了大量的典型题目,从而能够提高考生的应试能力。

本书是专为参加本门课程自学考试的考生编写的,是一本具有实用性的考前辅导用书。另外,本书也可作为同层次的其他学生的学习参考书。

### 图书在版编目(CIP)数据

数据结构导论自考应试指导/邹华跃主编. —南京:南京大学出版社,2000.8

(全国高等教育自学考试应试指导丛书)

ISBN 7-305-01619-5

I. 数... II. 邹... III. 数据结构-高等教育-自学考试-自学参考资料 IV. TP311.12

中国版本图书馆 CIP 数据核字(2000)第 69013 号

书 名 数据结构导论自考应试指导

主 编 邹华跃

副 主 编 王淮亭

责任编辑 王 勇

出版发行 南京大学出版社

地 址 南京汉口路22号 邮编 210093 电话 025-3593695

印 刷 合肥学苑印刷厂

经 销 全国各地新华书店

开 本 787×1092 1/16 印 张 11.25 字 数 281千字

版 次 2000年9月第1版 2000年9月第1次印刷

定 价 18.00元

ISBN 7-305-01619-5/TP·198

声明:(1)版权所有,侵权必究。

(2)本版书若有质量问题,可向经销商调换。

## 组编前言

国家教育部考试中心于2000年开始,正式执行自学考试新计划,同时使用新编的大纲和教材。

为适应新调整的考试计划及密切配合新大纲新教材开展助学辅导,中国计算机函授学院利用多年积累的自考教学辅导资源和经验,全面系统地剖析了本专业各门专业课程新大纲和教材的内容体系,重新组织编写了一套“全国高等教育自学考试计算机及应用专业应试指导”丛书,推向全国,以满足考生之急需,适应社会之需要。

这套丛书堪称“通关必读”,其主要特征是:

首先,担纲编写应试指导丛书的作者基本上都是该专业全国自考指定教材及大纲的主编。

其次,自考应试指导丛书的作者,都在书中融入了自己多年从事该专业自考教学辅导的直接经验。他们既是本专业的教授,又是自考辅导的专家,二者集于一身,有些作者就是当年在中央电视台担任自考辅导教学讲座的教授。

最后,精心组织、细心筹划、用心编撰,是这套丛书的又一质量保证。

编写该套丛书的指导思想是,切实解决考生自学应试中的三个问题:

(1)在自学过程中起到答疑解惑作用,帮助考生顺利阅读、掌握教材内容;

(2)帮助考生抓住课程重点、难点,不入迷津;

(3)帮助考生理清课程主线,建立清晰的知识结构体系,在掌握知识点的前提下,沉着应战,顺利过关。

较之其他专业而言,计算机及应用专业自学考试是有一定难度的,因此,请一位好“教师”,找一位好“辅导”,尤为重要。这套“自学考试指导”丛书,可望成为你攻克一门又一门课程,克服一个又一个难关的良师益友;帮助你扫清学习中的障碍,增强你的必胜信心,伴随你走向成功的彼岸。

我们真诚地为计算机及应用专业的广大考生奉献这份精品、真品。愿广大考生早成夙愿。

2000年1月

## 编者的话

《数据结构导论》是计算机及应用专业的一门专业技术基础课程,在计算机学科中起着承前启后的作用,在计算机技术的各个领域中也有着广泛的应用,因而是一门重要的课程。

数据结构涉及数据的组织、存储以及运算的一般方法,其原理及算法较为抽象,对刚刚接触计算机学科的考生来讲,了解与掌握其中的原理显得尤为困难;有些章节的内容自学起来难度特别大,犹如读天书一样;在解答习题时,往往也感到无从下手。作者借编写本书的机会,对多年来的教学实践进行了系统的总结,将自己对课程的认识、体会、感想融入到本书中,并尽可能从考生的角度出发,对难以理解的原理及算法进行了通俗化处理,以求高度概括,易学易懂。我们的愿望是,通过对本书的阅读,能让考生把握本课程的主线,加深对基本概念的理解,掌握求解数据结构问题的思路与方法,提高分析与解决问题的能力,直至最终顺利通过自学考试。

本书按照全国高等教育自学考试委员会电子电工及信息类专业委员会审定的《数据结构导论自学考试大纲》的要求,以考核知识点与考核要求为主线,再结合作者多年来的教学经验,从整体上对本书的结构及内容进行规划、编排,每章分为三个部分:知识点、典型题分析与解答、习题分析与解答。其中,知识点部分最能体现作者对本课程的认识与理解,是作者多年来教学工作的结晶,知识点部分的特色是通过简洁、通俗的语言对难以理解的原理、算法进行了归纳与总结;典型题分析与解答部分反映了作者对自考大纲的准确把握;我们不注重题量的多少,但是在题目的典型性、题型、难易程度等方面,力求与自考大纲贴近,以增强考生的应试能力;习题分析与解答部分对指定教材上的所有习题进行了详细的分析与解答,使考生能够知其然也知其所以然。这三部分内容各有特色,相互配合,相互补充。最后提供的三套模拟试卷,用于考生临考前的自查自检。在组卷时充分参考了本课程的历年自考试卷,以便大家能从中体会到自考试卷的题型、题量、考核点及试题的难易程度。由此,我们有理由相信,本书能够成为考生的良师益友,是考前辅导的好帮手。

本书在编写过程中得到了中国计算机函授学院编写中心牛允鹏教授及其他领导、同志们具体指导与大力支持;合肥工业大学计算机工程学院的胡学钢博士对书稿进行了认真仔细的审阅,并提出了宝贵的意见和建议;中国计算机函授学院图书编辑部及激光照排室的同志也为本书的出版付出了辛勤的劳动。谨向以上同志表示衷心的感谢及崇高的敬意。

由于本书涉及的题量较大,在分析、解答上可能会存在不够完整或疏漏之处,恳请广大读者批评指正。

编者

2000年6月

# 目 录

第一部分 知识点与典型题解	(1)
第1章 概 论	(2)
1.1 知识点	(2)
1.2 典型题分析与解答	(5)
1.3 课后习题分析与解答	(7)
第2章 线性表	(9)
2.1 知识点	(9)
2.2 典型题分析与解答	(15)
2.3 课后习题分析与解答	(26)
第3章 栈、队列和数组	(39)
3.1 知识点	(39)
3.2 典型题分析与解答	(44)
3.3 课后习题分析与解答	(49)
第4章 树	(58)
4.1 知识点	(58)
4.2 典型题分析与解答	(64)
4.3 课后习题分析与解答	(76)
第5章 图	(92)
5.1 知识点	(92)
5.2 典型题分析与解答	(97)
5.3 课后习题分析与解答	(102)
第6章 查找表	(110)
6.1 知识点	(110)
6.2 典型题分析与解答	(113)
6.3 课后习题分析与解答	(114)
第7章 文件	(122)

7.1 知识点 .....	(122)
7.2 典型题分析与解答 .....	(123)
7.3 课后习题分析与解答 .....	(125)
第8章 排序 .....	(129)
8.1 知识点 .....	(129)
8.2 典型题分析与解答 .....	(132)
8.3 课后习题分析与解答 .....	(136)
第二部分 模拟试卷分析与解答 .....	(145)
数据结构导论模拟试卷(一) .....	(146)
数据结构导论模拟试卷(一)分析与解答 .....	(150)
数据结构导论模拟试卷(二) .....	(156)
数据结构导论模拟试卷(二)分析与解答 .....	(159)
数据结构导论模拟试卷(三) .....	(165)
数据结构导论模拟试卷(三)分析与解答 .....	(169)

---

# 第一部分

---

## 知识点与典型题解

“内容概要”高度概括每章的考核内容,以帮助考生有的放矢地去掌握有关内容;“典型题解”是具有代表性的题,书中通过“分析”和“解答”具有“代表性”的试题形式,反映出自学考试数据结构导论试题的深浅度,有利于考生把握尺度,顺利过关。在本部分中,将按数据结构导论考试大纲来组织相关内容,分成以下八个章节:

- 第1章 概论
- 第2章 线性表
- 第3章 栈、队列和数组
- 第4章 树
- 第5章 图
- 第6章 查找表
- 第7章 文件
- 第8章 排序

在每一章中,都将围绕相关内容提炼出考核知识点,对每个知识点不像教材那样详细讲解,而是给出结论性的提示,这是本书最具特色的地方。考生只需掌握有关知识点,再结合书中典型题解,便可迅速了解相关内容的考题形式、深度、广度和难度。当然,同一问题可能会以不同题型出现,但这不过是一种命题技巧而已。

本书汇集了从历年数据结构导论试卷中挑选出的大量典型试题和笔者本人认为值得注意的问题。这些问题不敢说今后定然考到,但是基本上覆盖了数据结构导论自学考试的全部内容。



# 第 1 章 概 论

计算机是一种对信息进行加工处理的机器。研究计算机,主要是研究计算机信息的组织、加工和处理方法。因此,对计算机应用专业的学生来说,理解信息的组织、加工、处理等方面的基本概念显得尤其重要。本章讲述数据结构的概念、实现方法及算法的时间复杂性分析。

## 1.1 知 识 点

### 1. 数据、数据元素和数据项的概念

数据是指所有能输入到计算机中并能被计算机加工、处理的符号的集合。它是信息的载体,其含义极其广范,诸如数、符号、字体、图形、声音等都可以看作是数据。因此,在概念上不同于大家平常理解的“数”的概念。

数据元素是数据的基本单位,通常具有完整、确定的实际意义,并被当作运算的基本单位。比如数据库中的库文件可以被看作数据,而其中的一条记录就对应于一个个数据元素,它们有实际意义,也是数据库操作的基本单位。在后续章节中,数据元素又被分为元素、记录、结点、顶点,以便叙述上的方便、形象。

数据项是数据不可再分割的最小标识单位,它不具有完整的实际意义,通常仅反映数据元素某一方面的属性,在很多场合下,数据项又被称为字段、数据域。

综合起来看,数据、数据元素和数据项反映出了数据组织的三个层次,它们之间的关系是数据可由若干个数据元素构成的,而数据元素又可由若干个数据项构成。

### 2. 数据的逻辑结构

通常数据非常复杂,其中的数据元素之间可能存在着各种各样的关系。比如一个工厂里的工人之间,可能存在着上下级关系、血缘关系、同乡关系,同学关系等等,用数据结构的眼光来看,所有这些关系都可以抽象为数据元素之间的逻辑关系,而数据元素之间的逻辑关系的整体就构成了数据的逻辑结构。由此我们知道,数据的逻辑结构实际上就是数据的组织形式,它反映的是数据元素之间的一种关联方式或称“邻接关系”。

数据的逻辑结构分为四种基本类型:集合、线性结构、树形结构和图状结构。

#### (1) 集合

如果数据中的数据元素之间不存在任何逻辑关系,则称其逻辑结构为集合。也就是说,集合是指数据元素之间“没有关系的关系”的整体,因此集合的组织形式是最松散的,不受任

何制约。

### (2) 线性结构

在线性结构中,数据元素之间的关系是呈线性排列的。比如,百家姓中排列的赵、钱、孙、李、周、吴、郑、王……等姓之间就是一种线性关系。线性结构的特点是有头有尾,相互之间次序分明。

### (3) 树形结构

有些数据元素之间的关系呈现出一种层次特性,比如同事之间的上下级关系,家族中的血缘关系等等。从形态上看,这种层次关系像是一棵倒置的树,故把这种逻辑结构形象地称之为树形结构。树形结构的特点就是它的层次性。

### (4) 图状结构

图状结构是四种基本逻辑结构中最复杂的一种。因为在这种逻辑结构中,任意两个数据元素之间都有可能发生关联,而且这种关联没有任何规律可寻。比如城市之间的公路网或通讯网就是一种图状结构。

## ☞ 提个醒

关于逻辑结构,教材上特别指出的以下几点须引起注意:

- 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) 逻辑结构与所含数据元素的个数无关;
- 4) 逻辑结构与数据的存储无关,它是独立于计算机的。

## 3. 运算的概念

数据的运算就是指对数据施加的操作。最常用的运算有检索、插入、删除、更新、排序等。这些运算实际上是在抽象的数据上所施加的抽象操作。因此在这里,我们只关心这些操作是“做什么”的,而无须考虑“如何做”。

有一点需要明确,数据的运算是定义在数据的逻辑结构上的,因此每一种逻辑结构对应着一个运算的集合。

## 4. 数据结构的概念

通俗地讲,数据结构是带有关系的数据元素的集合。研究数据结构的目的是为了在计算机中实现对它们的操作,这样就引出两个问题:

第一,数据结构如何在计算机中表示?

第二,在数据结构上定义怎样的操作(运算)?

由此,我们就可以知道数据结构包含了三个方面的内容:

- 1) 数据的逻辑结构:它反映的是数据元素之间的逻辑关系;
- 2) 数据的存储结构:即数据结构在计算机内是如何表示的;
- 3) 数据的运算:定义在逻辑结构上的、对数据元素所施加的操作。

综上所述,数据结构的定义可以被描述为:由某种逻辑关系组织起来的一批数据,按一

定的存储表示方式把它们存储在计算机中,并在这些数据上定义了一个运算的集合。

## 5. 存储结构和运算实现

数据的存储结构就是指数据在计算机内的表示,它涉及数据元素的表示及元素之间关系的表示两个方面。数据的存储方式有四种,它们的基本思想及特征如下:

### (1) 顺序存储结构

这种存储方式是把逻辑上相邻的数据元素存储到物理位置上相邻的存储单元里。因此,数据元素之间的逻辑关系就由存储单元的相邻关系来体现。

### 提个醒

有两点需要注意:

1) 顺序存储结构使用的是一段连续空间,一般中间不允许有间隙;

2) 要“按一定的次序”将元素装入存储器里,这种次序同时也决定了对数据元素的存取方式。在非线性结构的顺序存储结构中,这一点表现得尤为突出。

### (2) 链式存储结构

在这种存储方式中,数据元素之间的逻辑关系由附加的指针来表示。因此,对数据元素的具体存放位置没有限制,只要在存放一个数据元素的同时存储一个或多个指针,让这些指针指向与本元素有关联的数据元素就可以了。

### (3) 索引存储结构

该方法是在存储数据元素的同时,建立一张附加的索引表,用索引表中的索引项来指示各数据元素的存储位置或存储位置的区间起点。

### (4) 散列存储方式

这种存储方式的本质是,将数据元素的关键字带到散列函数中,通过计算得到其存储地址。也就是说,它是通过散列函数在数据元素本身与其存储地址之间建立起直接的联系。

下面简单谈谈运算的实现。前面讲到运算是定义在数据的逻辑结构之上的,这时我们只关心它的功能,即能“做什么”,而运算的实现却是针对数据的存储结构而言的,它的核心问题是应该“怎样做”,即平常我们所说的算法设计。

## 6. 算法及其描述

通俗地讲,算法就是解决问题的方法和步骤,它是用语言来描述的。一个算法可以用自然语言、标准程序设计语言或伪语言来描述。用自然语言来描述算法非常随意、方便,但是它有一个致命的弱点:表示不严格,会出现二义性。标准程序设计语言正好和它相反,表示很严格,不会出现二义性,但是语言规范太死,必须经过培训才会运用。综合上述两类的优缺点,人们想到用一种类语言(又称为伪语言,假语言)来描述算法,使它既能自然、方便的被使用,又具有一定的严格性,不会出现二义性。现在常用的类语言有类 C 语言、类 PASCAL 语言。

## 7. 算法分析

因为求解同一问题可以有许多不同的算法,那么怎样来衡量算法的好坏呢?这就涉及

到算法分析。衡量一个算法是否优异,主要取决于以下四点:

- 1) 能否正确地实现预定的功能;
- 2) 执行算法所消耗的时间;
- 3) 执行算法所消耗的存储空间;
- 4) 算法是否易于理解,易于编码,易于调试等等。

在本教材中,主要讨论的是算法的时间性能,偶尔也讨论空间性能。一个算法所消耗的时间,与算法中每条语句执行的次数(也称为频度)有直接关系。在很多情况下,语句的执行次数又取决于初始数据量  $n$ (称其为问题的规模)的多少,而一个算法的时间复杂性就是该算法所求解问题规模  $n$  的函数。当问题规模  $n$  趋向于无穷大时,则把时间复杂性的数量级称为算法的渐近时间复杂性。在算法分析时,经常将渐近时间复杂性简称为时间复杂性,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。

在很多算法中,其时间复杂性还与所处理数据的分布状态有关。有时会根据各种可能出现的数据分布状态中最坏的情况来估计算法的最坏时间复杂性;有时也会对数据分布作出某种假设(如等概率),然后估计算法的平均时间复杂性。


### 提个醒

上述关于数据分析涉及到五个方面,而后续章节中对每一种数据结构的讨论,也是按这五个方面(步骤)来展开的:

- 1) 逻辑结构上的特点;
- 2) 定义在逻辑结构上的基本运算;
- 3) 数据在计算机内的表示(即存储结构);
- 4) 在具体存储结构上的运算实现;
- 5) 对算法的时间性能、空间性能进行评价。

每种数据结构在上述五个方面都是密切相关的,而不同数据结构的对应方面也有着相互联系。所以在自学过程中,大家要善于进行比较,找出彼此间的相同点和不同点,这样有助于加深理解并逐渐在头脑中形成一个完整的体系。

## 1.2 典型题分析与解答

 通常要求同一逻辑结构中的所有数据元素具有相同的特性,这意味着( )。

- A) 数据元素具有同一特点。
- B) 不仅数据元素所包含的数据项的个数要相同,而且对应数据项的类型要一致。
- C) 每个数据元素都一样。
- D) 数据元素所包含的数据项的个数要相等。

**【分析】**同一逻辑结构中的所有数据元素必须具有相同的数据域,而且对应数据项的类型要一致。因为一方面只有同一类型的数据元素才能归类到同一种结构中;另一方面,在

计算机存储器中表示数据元素时,还必须为它们定义相同的数据类型。故 B)答案是正确的。

【答案】B)。

2) 所谓数据的逻辑结构指的是数据元素之间的\_\_\_\_\_。

【分析】这道题容易出错的地方在于许多考生仅填写“逻辑关系”,这是不全面的,应该是“逻辑关系的整体”。

【答案】逻辑关系的整体。

3) \_\_\_\_\_与数据元素本身的内容和形式无关。

【答案】数据的逻辑结构。

4) 根据描述算法的语言不同,算法通常可以分为三类,它们分别是\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。

【答案】运行终止的程序可执行部分,伪语言算法,非形式算法。

5) 数据结构是相互之间存在一种或多种特定关系的数据元素的集合,它包括三方面的内容,分别是\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。

【答案】数据的逻辑结构,数据的存储结构,运算。

6) 从逻辑关系上讲,数据结构主要分为两大类,它们是\_\_\_\_\_和\_\_\_\_\_。

【答案】线性结构,非线性结构。

7) 下列程序段的时间复杂度是多少?

```
    :  
    for (i = 1; i <= n; i + + )  
    |  
    |   k + + ; for(j = 1; j <= n; j + + ) l + = k;  
    |  
    :
```

【分析】确切地计算一个算法的时间消耗没有任何实际意义。因为在具体问题中,问题规模  $n$  的大小是变化的,而且每台计算机的主频也不相同,所以我们关注的是算法中每条语句的执行次数(频度)。对上述程序段,语句执行的总次数为:

$$n + 1 + n + n(n + 1) + n^2 = 2n^2 + 3n + 1。$$

令  $T(n) = 2n^2 + 3n + 1,$

$T(n)$ 就能反映出程序段的时间消耗。当  $n$  较大时, $T(n)$ 的大小主要取决于  $n$  的数量级。借助数学上极限的概念,如果:

$$\lim_{n \rightarrow \infty} \frac{T(n)}{f(n)} = \text{常数},$$

表示当  $n \rightarrow \infty$  时, $T(n)$ 和  $f(n)$ 具有相同的数量级,称  $T(n)$ 与  $f(n)$ 同阶。

在本例里,假设  $f(n) = n^2$ ,则有

$$\lim_{n \rightarrow \infty} \frac{T(n)}{f(n)} = \lim_{n \rightarrow \infty} \frac{2n^2 + 3n + 1}{n^2} = 2,$$

即  $T(n)$  与  $n^2$  同阶。采用  $O$  记法, 有:  $T(n) = O(f(n)) = O(n^2)$

其中:  $T(n)$  就是算法的渐近时间复杂度, 简称为时间复杂度。而  $f(n)$  恰为算法中执行频度最大的那条语句的频度。由此得到求解时间复杂度的方法: 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。

**【答案】** $O(n^2)$ 。

**8** 分析下列程序段的时间复杂度。

```

:
i = 1;
while(i <= n)
    i = i * 2;
:

```

**【分析】**方法同上, 此处循环体里面是  $i = i * 2$ , 即每循环一次  $i$  值增加一倍, 所以执行次数与  $n$  之间是以 2 为底的对数关系, 故时间复杂度为  $O(\log_2 n)$ 。

**【答案】** $O(\log_2 n)$ 。

## 1.3 课后习题分析与解答<sup>①</sup>

### 1.1

(1)

**【答案】**数据 凡能被计算机存储、加工的对象通称为数据。

数据元素 它是数据的基本单位, 在程序中通常作为一个整体而加以考虑和处理。

区别 一个数据可以由一个或多个数据元素构成。

(2)

**【答案】**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”, 数据元素不可能是孤立存在的, 它们之间总是存在着某种关系, 这种数据元素之间逻辑关系的整体称为逻辑结构。一些表面上很不相同的数据可以有相同的逻辑结构, 逻辑结构是数据组织的某种“本质性”的东西, 所以逻辑关系是数据内部组织的主要方面。

数据的逻辑结构还有以下几点特别需要注意:

- 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。

(3)

**【答案】**逻辑结构反映数据元素之间的逻辑关系, 而存储结构是数据结构在计算机中的表示,

<sup>①</sup> 注: 此部分为自考教材中相应章节后的部分习题解答, 题号对应于教材中的题号。因是部分题解, 故题号不一定连续, 请读者注意。在本书的其他章节也有类似情况, 不再解释。

它包括数据元素的表示及其关系的表示。

(4)

**【答案】**运算是指在逻辑结构上施加的操作,而运算的实现是指一个完成该运算功能的程序。

相同点 运算和运算的实现都能完成对数据的“处理”或某种特定的操作。

不同点 运算只是描述处理功能,不包括处理步骤和方法,而运算的实现的的核心则是处理步骤。

1.2

(1)

**【答案】**int locate ( datatype a[1..n]; datatype k)

```
    { i = n;                               /* 搜索的起点 */
      while( i >= 1 && a[i] != k)
        i -- ;                             /* 往前搜索 */
      return(i);
    }
```

当查找不成功时,总是比较  $n+1$  次,所以最坏时间复杂性为  $n+1$ ,其量级为  $T(n) = O(n)$ 。

(2)

**【答案】**Void max\_smax (datatype a[1..n], max, smax)

```
    { max = a[1];smax = a[1];             /* max,smax 分别表示当前最大值和次最大值 */
      for(i = 2; i <= n; i + +)
        if(a[i] > max)
          { smax = max;
            max = a[i];                   /* 替换最大值 */
          }
        else if (a[i] > smax) smax = a[i]; /* 替换次最大值 */
    }
```

为了得到最大值和最小值,必须经过  $n-1$  次循环比较,所以最坏时间复杂性为  $n-1$ ,其量级为  $T(n) = O(n)$ 。

# 第 2 章 线 性 表

线性表是最简单、最常用的一种数据结构。本章的基本内容是:线性表的逻辑结构定义、各种存储结构的描述方法以及在线性表的顺序存储结构和链式存储结构上是如何实现基本运算的。

## 2.1 知 识 点

### 1. 线性结构的概念

线性结构是由  $n(n \geq 0)$  个结点(数据元素) $a_1, a_2, \dots, a_n$  组成的有限序列。其中  $a_1$  是有限序列的第一个元素,称其为起始结点, $a_n$  是有限序列的最后一个元素,称其为终端结点。对任意两个相邻的结点  $a_i, a_{i+1} (1 \leq i < n)$ , 由于  $a_i$  位于  $a_{i+1}$  前面,并且两者相邻,故我们称  $a_i$  是  $a_{i+1}$  的直接前趋。同理,称  $a_{i+1}$  是  $a_i$  的直接后继。

由上述定义,可以看出线性结构(非空的线性结构)的基本特征是:

- 1) 有且仅有一个起始结点  $a_1$ , 它没有直接前趋, 只有一个直接后继  $a_2$ ;
- 2) 有且仅有一个终端结点  $a_n$ , 它没有直接后继, 只有一个直接前趋  $a_{n-1}$ ;
- 3) 其余结点  $a_i (2 \leq i \leq n-1)$  都有且仅有一个直接前趋  $a_{i-1}$  和一个直接后继  $a_{i+1}$ 。由此我们得到,线性结构中结点间的邻接关系实际上就反映出了结点之间的逻辑关系。

### 2. 线性表的概念

从逻辑上讲,线性表是一种线性结构。其中所含结点的个数  $n$  称为线性表的长度。

当  $n=0$  时,线性表为空表,记为  $( )$  或  $\emptyset$ ;

当  $n \neq 0$  时,线性表非空,记为  $(a_1, a_2, \dots, a_n)$ 。

对于线性表,常见的基本算法有以下几种:

- 1) INITATE(L) 对线性表 L 进行初始化,使其成为一个空表  $L = \emptyset$ 。
- 2) LENGTH(L) 求线性表 L 的长度(即所含结点的个数)。
- 3) GET(L, i) 当  $1 \leq i \leq n$  时,读取线性表 L 的第  $i$  个元素  $a_i$ 。
- 4) LOCATE(L, X) 在线性表 L 中查找值为 X 的结点。如果查找成功,运算结果为首次找到值为 X 的结点的序号,如果查找失败,运算结果为 0。
- 5) INSERT(L, X, i) 在线性表 L 的第  $i (1 \leq i \leq n+1)$  个位置上插入值为 X 的新结点,使编号为  $i, i+1, i+2, \dots, n$  的结点分别变为编号为  $i+1, i+2, \dots, n+1$  的结点。
- 6) DELETE(L, i) 删除线性表 L 的第  $i (1 \leq i \leq n)$  个结点,使编号为  $i+1, i+2, \dots, n$  的结



点分别变成编号为  $i, i+1, i+2, \dots, n-1$  的结点。

### 3. 线性表的顺序存储结构——顺序表

线性表最简单、最自然的存储方式是顺序存储方式,它是把线性表中的结点按逻辑次序依次存放到一组地址连续的存储单元里。它的特点是逻辑相邻的结点,其物理位置一定相邻。用这种方法存储的线性表简称为顺序表。

由于高级语言中的向量(一维数组)也是采用顺序存储方式来表示的,故可以用向量这种数据类型来描述顺序表。教材中给出了顺序表的类型定义描述,从中我们知道顺序表由两部分组成,一部分是用于存放结点的一个一维数组  $data$ ,另一部分是用于指示顺序表长度的变量  $last$ 。

假设顺序表  $L$  是一个  $sqliist$  类型的变量,那么可以通过下列方式对两个域进行访问:

$L.data[i]$  里存放的是顺序表中第  $i+1$  个结点  $a_{i+1}$ ,其中  $0 \leq i \leq n-1$ 。

$L.last$  既可以用于表示顺序表的长度,又可以用于指示终端结点  $a_n$  的存放位置( $a_n$  的存放单元序号为  $L.last-1$ )。

当顺序表中每个结点占用  $l(l \geq 1)$  个存储单元,而且已知起始结点  $a_1$  的第一个单元的地址是  $b$ ,则可以通过下列公式求得任一结点  $a_i(1 \leq i \leq n)$  的第一个存储单元的地址  $Loc(a_i)$ :

$$Loc(a_i) = b + (i - 1) * l$$

也就是说,在顺序表中,每个结点  $a_i$  的存储地址是该结点在线性表中的位置  $i$  的线性函数,只要知道  $a_1$  的起始地址和每个结点的大小,就可在相同时间内求出任一结点的存储地址,故顺序表是一种随机存取结构。

### 4. 插入、删除和定位运算在顺序表上的实现

#### (1) 插入运算

在顺序表中实现插入运算,其核心步骤是从  $a_n, a_{n-1}, \dots, a_i$  依次向后移一个位置,腾出第  $i$  个位置以便存放待插入的结点。但是在移动之前要判断一下当前顺序表是否已满、给定的插入位置是否合法。因此,教材上的  $insert\_sqliist$  算法由五个基本操作步骤组成:

- 1) 判断  $L$  是否已满? 当  $L.last = maxsize$  时,表示  $L$  已满;
- 2) 判断  $i$  值是否合法? 当  $i < 1$  或  $i > L.last$  时, $i$  值不合法;
- 3)  $a_n \sim a_i$  依次向后移一个位置;
- 4) 将  $x$  存放到第  $i$  个位置处;
- 5) 表长加 1。

#### (2) 删除运算

在顺序表中实现删除运算,其核心步骤是从  $a_{i+1} \sim a_n$  依次向前移一个位置,以使执行删除操作后的顺序表仍然是连续的。但是此处只需判断删除位置  $i$  是否合法,而无须判断当前顺序表是否为空。对照教材上的  $delete\_sqliist$  算法,它的三个基本操作步骤是:

- 1) 判断  $i$  值是否合法? 当  $i < 1$  或  $i > L.last$  时, $i$  值不合法;
- 2)  $a_{i+1} \sim a_n$  依次向前移一个位置;
- 3) 表长减 1。