总策划：秦洪涛

2006 考试专用

全国计算机等级考试
指定教材配套辅导

上机题库：资深专家设计
解题分析：每题均有权威专家为你详尽解析
视频课程：只针对二级公共基础知识（会员卡）
专家解疑：全国免费电话：800-810-0480
（只针对购书购卡的会员使用）

购买本书者 +5 获得全国计算机等级考试网（www.ncre.cn）价值30元上机模拟卡

全国计算机等级考试命题研究组 编

全国计算机等级考试

应试指导及模拟试题集

——二级公共基础知识+C语言程序设计(2006年版)

中国大地出版社

全国计算机等级考试丛书

应试指导及模拟试题集

二级公共基础知识＋C语言程序设计

全国计算机等级考试命题研究组　　编

中国大地出版社

内容简介

本书是由全国计算机等级考试命题研究组专家编写。教育部考试中心指定教材的同步配套指导，本书紧扣2004年教育部考试中心最新考试大纲编写，应试导向准确，针对性强。本书的试题经过精心设计，题型标准，考生只需用少量时间，通过实战练习，就能在较短时间内巩固所学知识，掌握要点、突破难点、把握考点、熟练掌握答题方法及技巧，适应考试氛围，顺利通过考试。

图书在版编目(CIP)数据

二级C语言程序设计应试指导及模拟试题集/全国计算机等级考试命题研究组编. —北京：中国大地出版社，2003.5

(全国计算机等级考试辅导丛书)
ISBN 7-80097-564-9

Ⅰ.二Ⅱ.全…Ⅲ.C语言-程序设计-水平考试-自学参考资料 Ⅳ.TP312

中国版本图书馆CIP数据核字(2003)第029956号

丛 书 名：全国计算机等级考试应试指导及模拟试题集系列
书　　名：二级C语言程序设计应试指导及模拟试题集
责任编辑：张　雄
出版发行：中国大地出版社
社址邮编：北京市海淀区大柳树路19号　100081
印　　刷：铁十六局印刷厂
开　　本：880×1230
印　　张：150
字　　数：5000千字
版　　次：2005年10月第1版
印　　刷：2005年10月北京第1次印刷
书　　号：ISBN 7-80097-564-9/TP·8
定　　价：300.00元(全套)

前 言

在信息时代，计算机与软件技术日新月异，发展迅猛，渗透到了经济、文化和社会的各个领域，迅速地改变着人们的观念、生活和社会结构。因此，计算机知识的掌握及应用毋庸置疑成了培养新型人才的一个重要环节。

国家教育部考试中心顺应社会发展的需要，于1994年推出“全国计算机等级考试”（简称NCRE），其目的是以考促学，向社会推广普及计算机知识，为选拔人才提供统一、公正、客观和科学的标准。1994年是推出计算机等级考试的第一年，当年参加考试的有1万余人；到2003年，报考人数已达251万余人。截止至2005年底，全国计算机等级考试共开考22次，考生人数累计超过1450万人，其中，有550多万考生获得了不同级别的证书。这充分证明该项考试适应了国家信息化发展的迫切需要，对计算机应用知识与技能的普及起到了有力的促进作用，成为了面向未来、面向新世纪培训人才、继续教育的一种有效途径。

参加NCRE的许多人都普遍感到这种考试与传统考试不同，除指定的教材外，缺少关于上机指导、笔试指导以及模拟试题方面的资料，因此，为配合社会各类人员参加考试，能顺利通过“全国计算机等级考试”，我们组织多年从事辅导计算机等级考试的专家在对近几年的考试深刻分析、研究基础上，并依据教育部考试中心最新考试大纲的要求，编写出这套指导应考者参加考试的备考辅导资料，本套丛书具有以下特点：

一、本套丛书自2000年在中国大地出版社出版以来，其后是不断修订再版，无论是内容还是题型，均以**教育部考试中心最新考试大纲**为纲，围绕**考生需求**为领，不断的作出修订和改进，力求把**轻略图书**做到最好。

二、在图书内容上，每本书均提供了**考试大纲**、**考试要求**、**知识重点**、**精典例题解析**、**命题规律预测**（提供了大量的反馈测试题）、最新**考试真题及答案**、**全真模拟试题**（含**笔试**、**上机**两部分），书中重点、难点明确，应试导向准确，试题经过精心设计，题型标准、针对性强。

三、本书采用**小5号字紧缩式**排版，每一页比同类其他书内容更充实、丰富，目的是让考生在同等硬件条件下汲取更多营养。

四、参与本书的编写者都为北京大学、清华大学等计算机专业人才，均是具有丰富教学和研究经验的专家、教授。另外，在此书的出版过程中，曾得到**全国计算机等级考试委员会顾问组组长罗晓沛教授**的悉心指导和热情支持，在此表示特别感谢。

五、凡购买本套丛书的读者，均可免费成为“轻略读者俱乐部”的会员。并享受购书带来的诸多实惠，欢迎读者积极参与。

六、本系列丛书和全国计算机等级考试（http://www.ncre.cn，该网站是隶属于教育部考试中心的官方网站，是全国计算机等级考试惟一权威信息发布网站）合作，只要您花5元钱就可以得到面值30元的上机考试卡。读者可以凭借该卡登录全国计算机等级考试网，注册成为该网会员，学习全国计算机等级考试网上课程，该课程提供全真上机考试模拟环境，汇集正式考试的各种试题、答案及答题技巧，练习、自测模式任选，随机抽题，熟悉上机考试环境，轻松过级不再是梦。

七、由于本套丛书修订出版时间仓促，谬误之处在所难免，恳请广大读者能及时给予批评指正，以促进本套丛书质量的不断提高，谢谢！

全国计算机等级考试命题研究组

2005.北京

二级C语言程序设计考试大纲

一、公共基础知识

基本要求

1. 掌握算法的基本概念。
2. 掌握基本数据结构及其操作。
3. 掌握基本排序和查找算法。
4. 掌握逐步求精的结构化程序设计方法。
5. 掌握软件工程的基本方法，具有初步应用相关技术进行软件开发的能力。
6. 掌握数据库的基本知识，了解关系数据库的设计。

考试内容

(一)基本数据结构与算法

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. 顺序查找与二分法查找算法；基本排序算法(交换类排序，选择类排序，插入类排序)。

(二)程序设计基础

1. 程序设计方法与风格。
2. 结构化程序设计。
3. 面向对象的程序设计方法，对象，方法，属性及继承与多态性。

(三)软件工程基础

1. 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。
2. 结构化分析方法，数据流图，数据字典，软件需求规格说明书。
3. 结构化设计方法，总体设计与详细设计。
4. 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。
5. 程序的调试、静态调试与动态调试。

(四)数据库设计基础

1. 数据库的基本概念：数据库，数据库管理系统，数据库系统。
2. 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。
3. 关系代数运算，包括集合运算及选择、投影、连接运算，数据库规范化理论。
4. 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。

考试方式

1. 公共基础知识的考试方式为笔试，与C语言程序设计(C++语言程序设计、Java语言程序设计、Visual Basic语言程序设计、Visual FoxPro数据库程序设计或Access数据库程序设计)的笔试部分合为一张试卷。

公共基础知识部分占全卷的30分。

2. 公共基础知识有10道选择题和5道填空题。

二、C语言程序设计

基本要求

1. 熟悉TURBO C集成环境。

2. 熟练掌握结构化程序设计的方法,具有良好的程序设计风格。

3. 掌握程序设计中简单的数据结构和算法。

4. TURBO C的集成环境下,能够编写简单的C程序,并具有基本的纠错和调试程序的能力。

考试内容

(一)C语言的结构

1. 程序的构成,MAIN函数和其他函数。

2. 头文件、数据说明、函数的开始和结束标志。

3. 源程序的书写格式。

4. C语言的风格。

(二)数据类型及其运算

1. C的数据类型(基本类型、构造类型、指针类型、空类型)及其定义方法。

2. C运算符的种类、运算优先级和结合性。

3. 不同类型数据间的转换与运算。

4. C表达式类型(赋值表达式、算术表达式、关系表达式、逻辑表达式、条件表达式、逗号表达式)和求值规则。

(三)基本语句

1. 表达式语句,空语句和复合语句。

2. 数据的输入与输出,输入输出函数的调用。

3. 复合语句。

4. GOTO语句和语句标号的作用。

(四)选择结构程序设计

1. 用IF语句实现选择结构。

2. 用SWITCH语句实现多分支选择结构。

3. 选择结构的嵌套。

(五)循环结构程序设计

1. FOR循环结构。

2. WHILE和DO WHILE循环结构。

3. CONTINUE语句和BREAK语句。

4. 循环的嵌套。

(六)数组的定义和引用

1. 一维数组和多维数组的定义、初始化和引用。

2. 字符串与字符数组。

(七)函数

1. 库函数的正确调用。

2. 函数的定义方法。

3. 函数的类型和返回值。

4. 形式参数与实在参数,参数值的传递。

5. 函数的正确调用,嵌套调用,递归调用。

6. 局部变量和全局变量。

7. 变量的存储类别(自动、静态、寄存器、外部),变量的作用域和生存期。

8. 内部函数与外部函数。

(八)编译预处理

1. 宏定义:不带参数的宏定义;带参数的宏定义。

2. "文件包含"处理。

(九)指针

1. 指针与指针变量的概念,指针与地址运算符。

2. 变量、数组、字符串、函数、结构体的指针以及指向变量、数组、字符串、函数、结构体的指针变量。通过指针引用以上各类型数据。

3. 用指针作函数参数。

4. 返回指针值的指针函数。

5. 指针数组,指向指针的指针,MAIN 函数的命令行参数。

(十)结构体(即"结构")与共用体(即"联合")

1. 结构体和共用体类型数据的定义方法和引用方法。

2. 用指针和结构体构成链表,单向链表的建立、输出、删除与插入。

(十一)位运算

1. 位运算符的含义及使用。

2. 简单的位运算。

(十二)文件操作

只要求缓冲文件系统(即高级磁盘 I/O 系统),对非标准缓冲文件系统(即低级磁盘 I/O 系统)不要求。

1. 文件类型指针(FILE 类型指针)。

2. 文件的打开与关闭(FOPEN,FCLOSE)。

3. 文件的读写(FPUTC,FGETC,FPUTS,FGETS,FREAD,FWRITE,FPRINTF,FSCANF 函数),文件的定位(REWIND,FSEEK 函数)。

考试方式

1. 笔试:120 分钟,满分 100 分,其中含公共基础知识部分的 30 分。

2. 上机:60 分钟,满分 100 分。

目 录

第 1 部分　应试指导:考试大纲串讲

1.1　公共基础知识

1.1.1 算法的基本概念

本节要求理解算法、算法的时间复杂度和空间复杂度等概念;熟悉常用算法的时间复杂度次序和时间复杂度分析方法。

考核知识点(一):算法

算法是对特定问题求解步骤的描述,它是指令的有限序列。

考核知识点(二):算法五特性

1. 输入性:一个算法有 n(n≥0)个初始数据的输入;
2. 输出性:算法必须有一个或多个有效信息的输出,这些信息同其输入有着某种特定的关系;
3. 有穷性:算法中每一条指令的执行次数必须是有限的;
4. 确定性:算法中每条指令的含义必须明确,无二义性;
5. 可行性:算法中的操作都必须是可行的,即必须是能具体实现的基本操作。每条指令都应在有限时间内完成。

考核知识点(三):算法的基本要素

算法的基本要素通常有两种:

一是对数据对象的运算和操作;

二是算法的控制结构。

考核知识点(四):算法的控制结构

算法中各操作之间执行的顺序称为算法的控制结构。

考核知识点(五):算法的描述

1. 以文字框图进行图示的算法流程图;
2. 用自然语言描述的算法规则及其基本思想;
3. N-S 结构化流程图;
4. 体现结构化程序设计原则的类 Pascal 或类 C 语言等描述方式;
5. 在计算机实现时采用的 Pascal 或 C 等高级语言形式。

考核知识点(六):算法设计的基本方法

算法设计的基本方法常见的方法有:

1. 列举法;
2. 归纳法;
3. 递推;
4. 递归;
5. 减半递推技术;
6. 回溯法。

考核知识点(七):算法的优劣

算法的优劣与其正确性、可读性、可维护性、健壮性、时间复杂度、空间复杂度,以及实现的难易程度有关。

时间复杂度和空间复杂度是衡量算法优劣的关键指标。近年程序设计中越来越重视算法的可行性、可读性和可维护性。

考核知识点(八):时间复杂度

算法执行的效率与算法策略、问题规模、书写语言、编译效率，以及机器速度都有关系。算法本身对执行效率的影响主要取决于关键语句执行的次数。

时间复杂度通常分为平均算法时间复杂度和最坏时间复杂度。平均时间复杂度和最坏时间复杂度一般在同一数量级，最坏时间复杂度往往比平均时间复杂度更有实用价值。

考核知识点(九)：大O方法

一般情况下，算法中基本操作重复的次数是问题规模n的某个函数f(n)，算法的时间量度记作：

$T(n)=O(f(n))$

称作算法的渐进时间复杂度，简称时间复杂度。

该式含义为："若T(n)和f(n)是定义在正整数集合上的两个函数，则$T(n)=O(f(n))$表示存在正的常数C和n_0，使得当$n\geq n_0$时都满足$0\leq T(n)\leq C\cdot f(n)$。"

即这两个函数当整型自变量n趋向于无穷大时，两者的比值是一个不等于0的常数。

考核知识点(十)：常见时间复杂度次序

常见的时间复杂度的次序为：常数阶$O(1)$、对数阶$O(\log_2 n)$、线性阶$O(n)$、线性对数阶$O(n\log_2 n)$、平方阶$O(n^2)$、立方阶$O(n^3)$、k次方阶$O(n^k)$，以及指数阶$O(2^n)$。

考核知识点(十一)：算法的空间复杂度

算法的空间复杂度一般指算法所需的内存空间。

考核知识点(十二)：算法的选择

高效且低内存要求的算法最好，但二者一般总需要取舍折中。与此同时，可行性和可维护性也十分重要。如果一个算法难于程序设计，即使它有较高的效率也不足取。在该算法很少使用的时候尤其如此，因为编制、调试和维护的代价很高。但是多次使用的算法，例如在导弹跟踪目标技术中计算π的值，则应选择尽可能快的算法。

1.1.2 数据结构基本概念

本节主要解数据、数据元素、数据项、数据结构、逻辑结构、存储结构、线性结构，以及非线性结构。

考核知识点(一)：数据

数据就是指能够被计算机识别、存储和加工处理的信息的载体。

考核知识点(二)：数据元素

数据元素是数据的基本单位，有时一个数据元素可以由若干个数据项组成。

考核知识点(三)：数据项

数据项是具有独立含义的最小标识单位。有时，数据元素也被称为元素、结点、顶点或记录；数据项被称为字段或属性。

考核知识点(四)：数据结构

从集合论的观点出发，数据结构(DS)可形式地描述成：

$DS=(D,R)$

式中，D是数据元素的有限集合，R是D上关系的有限集合。这两者的有机结合，表述了DS的内在规律性。其中包括三方面的内容，即数据的逻辑结构、存储结构以及对数据的操作性。

考核知识点(五)：逻辑结构

数据结构有逻辑上的数据结构和物理上的数据结构之分。逻辑上的数据结构反映各数据之间的逻辑关系，它与数据的存储无关，独立于计算机。数据的逻辑结构分两大类：线性结构和非线性结构。数据的逻辑结构常被简称为数据结构。

考核知识点(六)：逻辑结构分类

逻辑结构有四种基本类型：集合结构、线性结构、树形结构和网状结构。

表和树是最常用的两种高效数据结构，许多高效的算法都可以用这两种数据结构来设计实现。

考核知识点(七)：存储结构

物理上的数据结构反映了各数据在计算机内部的实际存储安排。常用数据存储方法有四种：顺序存储方法、链表存储方法、索引存储方法和散列存储方法。

考核知识点(八)：线性结构

线性结构是数据逻辑结构中的一类，它的特征是若结构为非空集，则该结构有且只有一个开始结点和一个终端结点，并且所有结点都最多只有一个直接前趋和一个直接后继。线性表就是一个典型的线性结构。

考核知识点(九):非线性结构

非线性结构:数据逻辑结构中的另一大类,它的逻辑特征是一个结点可能有多个直接前趋和直接后继。

考核知识点(十):数据结构的优劣

如果一个数据结构可以通过某种"线性规则"被转化为线性的数据结构(例如线性表),则称它为好的数据结构。好的数据结构通常对应于好的(高效的)算法。这是由计算机的计算能力决定的,因为计算机本质上只能存取逻辑连续的内存单元,因此,如果没有线性化的结构,逻辑上是不可计算的。

1.1.3　线性表及其顺序存储结构

本节了解线性表的逻辑结构特征和基本实现方法,常见线性表的六种基本运算;掌握顺序表的特征、顺序表中结点地址的计算,掌握顺序表上实现的基本运算(算法)。

考核知识点(一):线性结构特点

1. 存在惟一一个"第一个"数据元素;
2. 存在惟一一个"最后一个"数据元素;
3. 除第一个外,集合中每个元素只有一个直接前驱;
4. 除最后一个外,集合中每个元素只有一个直接后继。

考核知识点(二):线性表

相同数据类型元素的有限序列叫线性表。如,$(a_1,a_2,\cdots,a_{n-1},a_n)$。其中,$a_1$ 为首元,a_n 为末元;a_i 的后继是 a_{i+1},$i=1,\cdots,n-1$。a_n 没有后继;a_i 的前驱是 a_{i-1},$i=2,\cdots,n$。a_1 没有前驱;a_i 可以是基本数据类型也可以是自定义类型。线性表是最简单最常用的数据结构。

考核知识点(三):线性表的长度

一个线性表中数据元素的个数($n\geq 0$)叫线性表的长度;没有数据的线性表叫空表,空表的长度 $n=0$;线性表是最简单的也是最基本的数据结构,可以用来构造字符串、集合、栈和队列,并且可以用于排序。

考核知识点(四):顺序表和链表

线性表可以顺序表示,用一组地址连续的存储单元一次存储数据元素。线性表也可以用线性链表表示。

顺序存储就是按线性表的逻辑结构次序依次存放在一组地址连续的存储单元中。存储单元中的各元素的物理位置和逻辑结构中各结点相邻关系是一致的。

链表是用一组任意的存储单元来存放线性表的结点,这组存储单元可以分布在内存的任何位置上。因此,链表中结点的逻辑次序和物理次序不一定相同。所以为了能正确表示结点间的逻辑关系,在存储每个结点值的同时,还须存储其后继结点的地址信息(如指针)。这两部分信息共同组成链表中的结点结构。

考核知识点(五):存储方式的选用(如表 1-1 所示)

表 1-1　存储方式的选用

	顺序表	链表
基于空间考虑	适于线性表长度变化不大,易于事先确定其大小时采用	适于当线性表长度变化大,难以估计其存储规模时采用
基于时间考虑	由于顺序表是一种随机存储结构,当对线性表的操作主要是查找时,宜采用	链表中对任何位置进行插入和删除都只需修改指针,所以以这类操作为主的线性表宜采用链表做存储结构。若插入和删除主要发生在表的首尾两端,则宜采用尾指针表示的单循环链表

考核知识点(六):表运算

线性表上定义的基本运算主要有构造空表、求表长、取结点、查找、插入、删除等。常见的运算如表 1-2 所示(L 代表一个线性表)。

表1-2 表的运算

运算	含义
L. end	为了方便，假定表L的结束元素 a_n 之后还有一个位置，用属性end的值来表示这个位置。该属性的值为TPosition类型
L. first	这是一个属性，其值为表L中第一个元素的位置。当L为空表时，值为L. end
L. last	这是一个属性，其值为表L中最后一个元素的位置。当L为空表时，值为L. end
L. length	返回表L的长度，即元素个数
L. isempty( )	如果表L为空表(长度为0)则返回true，否则返回false
L. next(p)	这是一个函数，函数值为表L中位置p的后继位置。如果p是L中结束元素的位置，则L. next(p) = L. end。当L中没有位置p或p = L. end时，该运算无定义
L. prev(p)	这是一个函数，函数值为表L中位置p的前驱位置。当L中没有位置p或p是L中开始元素的位置时，该运算无定义
L. get(p)	这是一个函数，函数值为L中位置p处的元素。当p = L. end或L中没有位置p时，该运算无定义
L. put(x,p)	该方法将元素x放置到表L的位置p处，如果位置p处已有元素，则用x取代该元素；如果p = L. end或者L中没有位置p，则该运算无定义
L. insert(x,p)	在表L的位置p处插入元素x，并将原来占据位置p的元素及其后面的元素都向后推移一个位置。例如，设L为 $a_1,a_2,\cdots,a_n$，那么在执行L. insert(x,p)后，表L变为 $a_1,a_2,\cdots,a_{p-1},x,a_p,\cdots,a_n$。若p为L. end，那么表L变为 $a_1,a_2,\cdots,a_n,x$。若表L中没有位置p，则该运算无定义
L. delete(p)	从表L中删除位置p处的元素。如，当L为 $a_1,a_2,\cdots,a_n$ 且执行L. delete(p)后，L变为 $a_1,a_2,\cdots,a_{p-1},a_{p+1},\cdots,a_n$。当L中没有位置p或p = L. end时，该运算无定义
L. locate(x)	这是一个函数，函数值为元素x在L中的位置。若x在L中重复出现多次，则函数值为x第一次出现的位置。当x不在L中时，函数值为L. end
L. makeempty	这是一个将L变为空表的方法
L. print	将表L中所有元素按位置的先后次序打印输出

考核知识点(七)：表的数组实现

表的数组实现是一种顺序存储实现方法。

用数组实现表时，将表类型TList定义为一个记录。它有两个域，第一个域是一个数组，用于存储表中的元素，数组的大小根据表可能达到的最大长度而定；第二个域是一个整型变量last，用于指出表中结束元素在数组中的位置。表中第i个元素($1 \leqslant i \leqslant last$)存储在数组的第i个单元中。

在这种情况下，位置变量的类型TPosition是整型，位置变量p表示数组的第p个单元即表中第p个元素的位置。End(L)的函数值为last + 1。

数组表的插入、删除和查找操作的算法时间复杂度均为O(n)。

1.1.4 栈和队列

本节了解栈的基本特征；熟悉顺序栈的特点；掌握栈的数组实现基本算法；简单了解栈的链表实现。了解队列的特点；熟悉循环队列的特点和基本实现；掌握循环队列的常用算法。

考核知识点(一)：栈的基本概念

栈是一种特殊的表，这种表只在表头进行插入和删除操作，如图1-1所示。因此，表头对于栈来说具有特殊的意义，称为栈顶。相应地，表尾称为栈底。不含有任何元素的栈称为空栈。

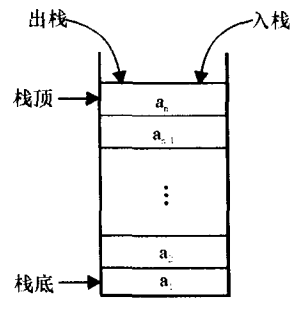

图1-1 栈示意图

数据对象:$D=\{\ a_i \mid a_i \in ElemSet,\ i=1,2,\cdots,n,\ n\geqslant 0\ \}$,

数据关系:$R1=\{\ <a_{i-1},\ a_i> \mid a_{i-1},\ a_i \in D,\ i=2,\cdots,n\ \}$

约定 a_n 端为栈顶,a_1 端为栈底。

假设一个栈 S 中的元素为 $a_n,a_{n-1},\cdots,a_1$,则称 a_1 为栈底元素,a_n 为栈顶元素。栈中的元素按 $a_1,a_2,\cdots,a_{n-1},a_n$ 的次序进栈。在任何时候,出栈的元素都是栈顶元素。即栈的修改是按后进先出的原则进行的。

因此,栈又称为后进先出(Last In First Out)表,简称为 LIFO 表。所以,只要问题满足 LIFO 原则,就可以使用栈。

考核知识点(二):栈的常见操作

常用的栈运算如表 1-3 所示。(S 为一栈)

表 1-3　栈运算

运　算	含　义
S. MakeNull( )	使 S 成为一个空栈
S. Top( )	这是一个函数,函数值为 S 中的栈顶元素
S. Pop( )	从栈 S 中删除栈顶元素,简称为出栈
S. Push(x)	在 S 的栈顶插入元素 x,简称为将元素 x 入栈
S. Empty( )	这是一个函数。当 S 为空栈时,函数值为 true,否则函数值为 false

栈的顺序存储可以使用数组实现。考虑到栈运算的特殊性,用一个数组 elements[1…maxlength]来表示一个栈时,将栈底固定在数组的底部,即 elements[1]为最早入栈的元素,并让栈向数组上方(下标增大的方向)扩展。同时,用一个游标 top 来指示当前栈顶元素所在的单元。当 top=0 时,表示这个栈为一个空栈。在一般情况下,elements 中的元素序列 elements[top],elements[top-1],…,elements[1]就构成了一个栈。

顺序栈 5 种基本操作的算法时间复杂度均为 O(1),如图 1-2 所示。

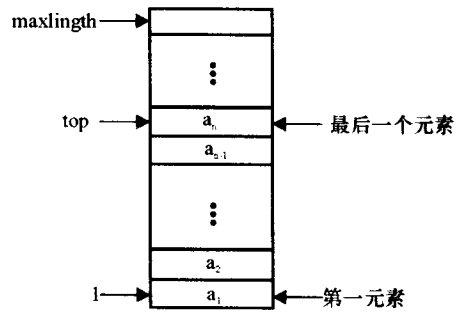

图 1-2　顺序栈

考核知识点(三):数组栈空间利用

在一些问题中,可能需要同时使用多个同类型的栈。为了使每个栈在算法运行过程中不会溢出,要为每个栈设置一个较大的栈空间。如果让多个栈共享同一个数组,动态地互相调剂,将会提高空间的利用率,并减少发生栈上溢的可能性。如图 1-3所示。

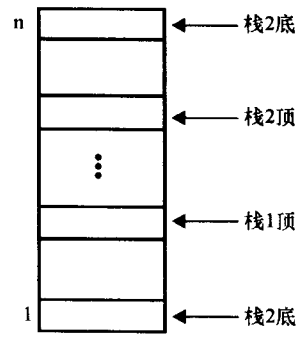

图 1-3　多个顺序栈共享同一个数组

考核知识点(四):链表栈

用链表作为栈的存储结构实现的栈也称为链栈。如图 1-4 所示。由于栈的插入和删除操作只在表头进行,因此用指针实现栈时没有必要像单链表那样设置一个表头单元。

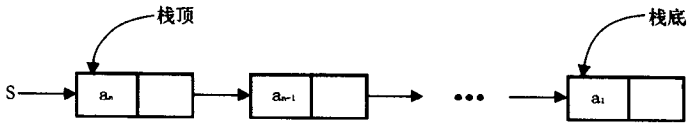

图 1-4　链栈

以上操作中只有 MakeNull 的时间复杂度为 O(n),其余的时间复杂度为 O(1)。

考核知识点(五):队列的基本概念

队列是一种特殊的线性表。如图 1-5 所示。对这种线性表,删除操作只在表头(称为队头)进行,插入操作只在表尾(称为队尾)进行。队列的修改是按先进先出的原则进行的,所以队列又称为先进先出(First In First Out)表,简称 FIFO 表。

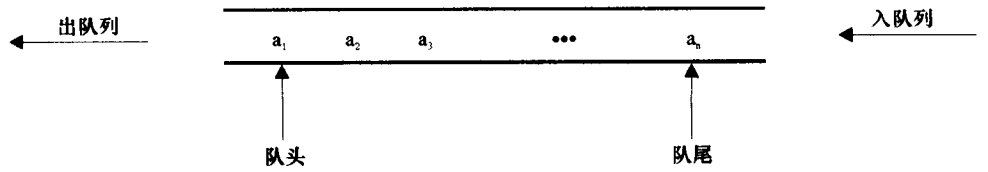

图 1－5　队列示意

数据对象：$D=\{a_i \mid a_i \in ElemSet,\ i=1,2,\cdots,n,\ n\geq 0\}$

数据关系：$R1=\{<a_{i-1},a_i> \mid a_{i-1},a_i \in D,\ i=2,\cdots,n\}$

约定其中 a_1 端为队列头，a_n 端为队列尾。

考核知识点(六)：队列的操作(如表 1－4 所示)

表 1－4　队列的操作(运算)

运　算	含　义
Front(Q)	这是一个函数，函数值返回队列 Q 的队头元素
Enqueue(x,Q)	将元素 x 插入队列 Q 的队尾。此运算也常简称为将元素 x 入队
Dequeue(Q)	将 Q 的队头元素删除，简称为出队
Empty(Q)	这是一个函数，若 Q 是一个空队列，则函数值为 true，否则为 false
MakeNull(Q)	使队列 Q 成为空队列

考核知识点(七)：循环队列

用一般数组实现队列效果并不好。

设想数组 Q[1…MaxLength]中的单元不是排成一行，而是围成一个圆环，即 Q[1]接在 Q[MaxLength]的后面。这种意义下的数组称为循环数组。如图 1－6 所示。

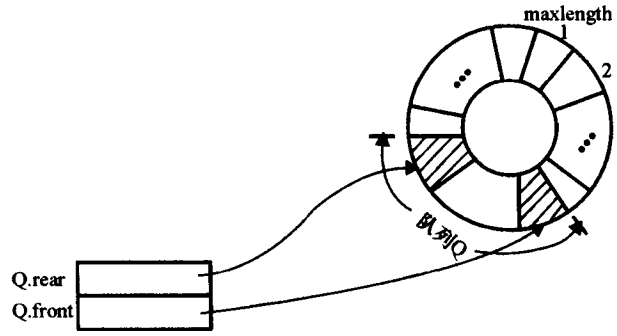

图 1－6　用循环数组实现队列

用循环数组实现队列时，将队列中从队头到队尾的元素按顺时针方向存放在循环数组里的一段连续的单元中。当需要将新元素入队时，可将队尾游标 Q. rear 按顺时针方向移一位，并在新的队尾游标指示的单元中存入新元素。出队操作也很简单，只要将队头游标 Q. front 依顺时针方向移一位即可。容易看出，用循环数组来实现队列可以在 O(1)时间内完成 Enqueue 和 Dequeue 运算。执行一系列的入队与出队运算，将使整个队列在循环数组中按顺时针方向移动。

考核知识点(八)：循环队列队空队满的表示

在循环数组中，不论用哪一种方式来指示队头与队尾元素，都要解决一个细节问题，即如何表示满队列和空队列。图 1－7给出一个例子，MaxLength＝6，队列中已有 3 个元素。用上述 3 种方法来指示队头和队尾元素，分别如图 1－7(a)、(b)和(c)所示。现在，有 3 个元素 a_4，a_5，a_6 相继入队，使队列呈“满”的状态，则如图 1－8 相应的(a)、(b)和(c)所示。

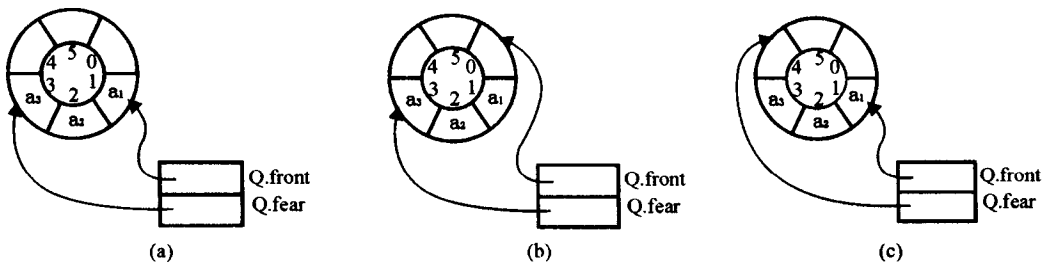

图 1－7　循环数中的队列

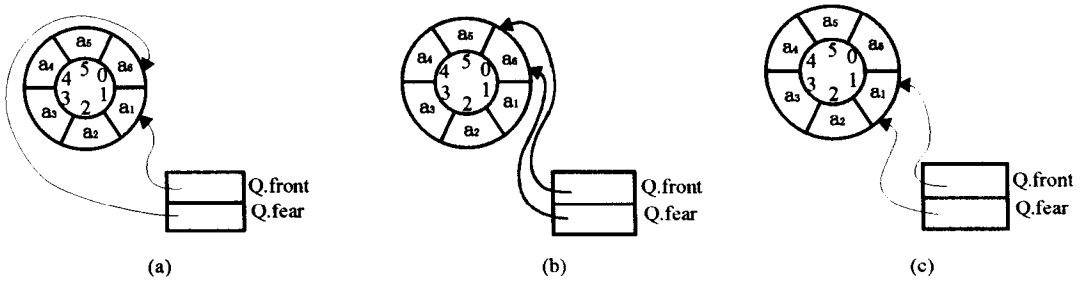

图 1 - 8　队列满的情形

如果在图 1 - 7 中,3 个元素 a_1,a_2,a_3 相继出队,使队列呈"空"的状态,则如图 1 - 9 相应的(a)、(b)和(c)所示。

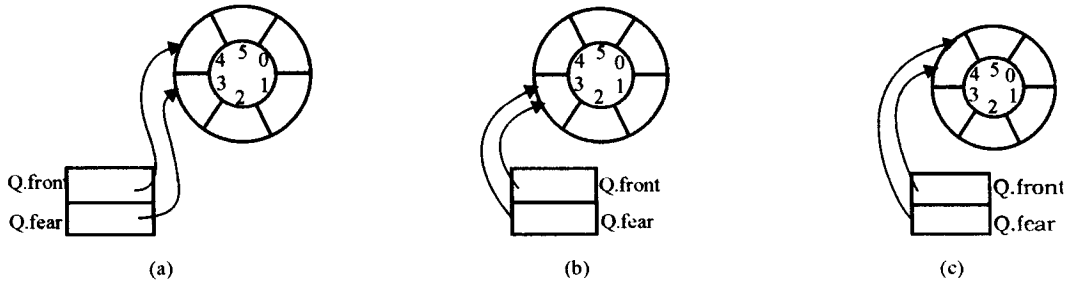

图 1 - 9　队列空的情形

比较图 1 - 8 和图 1 - 9 可以看到,不论采用哪一种方式指示队头和队尾元素,都需要附加说明或约定才能区分满队列和空队列。

通常有两种处理方法可以解决这个问题。其一是另设一个布尔量以注明队列是空还是满。其二是约定当循环数组中元素个数达到 MaxLength - 1 时队列为"满",使得队列满和队列空时的队头和队尾游标的相对位置不同,从而满队列和空队列得以区分。例如,在图 1 - 8 中,当元素 a_4 和 a_5 相继入队后,就使队列呈"满"的状态,如图 1 - 10 所示。比较图 1 - 9 和图 1 - 10 可知,只要测试队头和队尾游标的相对位置便可区分出满队列和空队列。

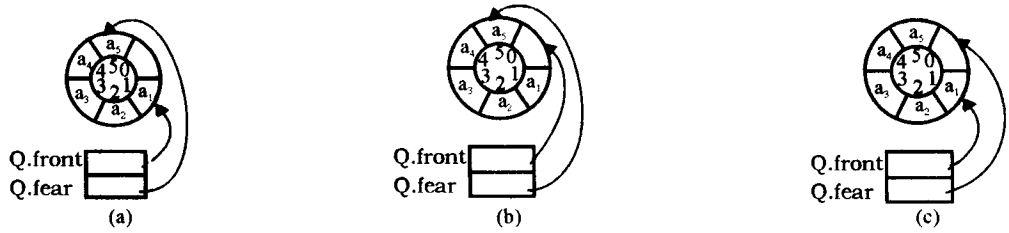

图 1 - 10　改进后的队列满的情形

考核知识点(九):循环队列的常见操作

队列的 5 种基本操作(如用循环队列实现),时间复杂度均为 O(1)。

考核知识点(十):链队列

用指针实现队列得到的是一个单链表。由于入队在队尾进行,所以用一个指针来指示队尾可以使入队操作不必从头到尾检查整个表,从而提高运算的效率。另外,指向队头的指针对于 Front 和 Dequeue 运算也是必要的。为了便于表示空队列,仍使用一个表头单元,将队头指针指向表头单元。当队头和队尾指针都指向表头单元时,表示队列为一个空队列。

其中 front 为队头指针,rear 为队尾指针。图 1 - 11 是用指针表示队列的示意图。

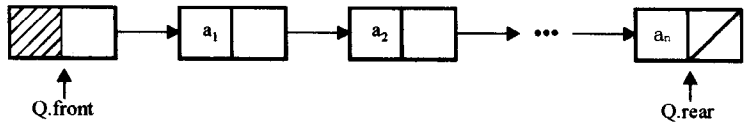

图1-11 链队列

考核知识点(十一):链队列的操作如表(1-5所示)

表1-5 链队列的操作(运算)

运算	含义	时间复杂度
Front(Q)	返回队列Q的队头元素	O(1)
Enqueue(x,Q)	将元素x插入队列Q的队尾	O(1)
Dequeue(Q)	将Q的队头元素删除	O(1)
Empty(Q)	检查Q是否为空	O(1)
MakeNull(Q)	使队列Q成为空队列	O(n)

1.1.5 线性链表

本节了解线性表链式存储的基本特征,单链表、循环链表和双向链表的基本概念;掌握单链表的实现和基本算法;熟悉循环链表和双向链表的基本特征。

考核知识点(一):线性表数组实现的优缺点如表1-6所示

表1-6 线性表数组实现的优缺点

优点是	缺点是
(1)无须为表示表中元素之间的逻辑关系增加额外的存储空间	(1)插入和删除运算不方便,除表尾的位置外,在表的其他位置上进行插入或删除操作都必须移动大量元素,其效率较低
(2)可以方便地随机访问表中任一位置的元素	(2)由于数组要求占用连续的存储空间,存储分配只能预先进行静态分配因此,当表长变化较大时,难以确定数组的合适大小,确定大了将造成浪费

考核知识点(二):线性链表

线性链表指线性表的链式存储结构。栈和队列都是操作受限的线性表因而可采用链式存储实现,这点已在前两节看到。

考核知识点(三):单链表

单链表中的每个元素包含一个数据域和一个指针域,其中的指针指向表中下一个元素所在的单元,最后一个元素单元中的指针为空指针nil。通常,还为每一个表设置一个表头单元header,其中的指针指向开始元素中所在的单元,但表头单元header中不含任何元素。设置表头单元的目的是为了使表运算中的一些边界条件更容易处理。这种用指针来表示表的结构通常称为单链接表,简称为单链表或链表。

表示空表的单链表只有一个单元,即表头单元header,其中的指针是空指针nil。

考核知识点(四):单链表的寻尾结点操作

若表的长度为n,按这样计算End(L)需要扫描整个链表,因此需要O(n)时间,效率很低。如果需要频繁地调用函数End,则可以采用下面的两种方法之一来提高效率。

(1)在链表结构中多设一个指向链表结束单元的表尾指针。对此,通过表尾指针就可以在O(1)时间内实现End(L)。

(2)尽可能避免调用End(L)。

考核知识点(五):单链表的插入操作

操作之前如图1-12所示。