



全国计算机等级考试 专家讲评

二级 C++ 语言程序设计

全国计算机等级考试专业网站 策划

李 季 黄志雄 等编著

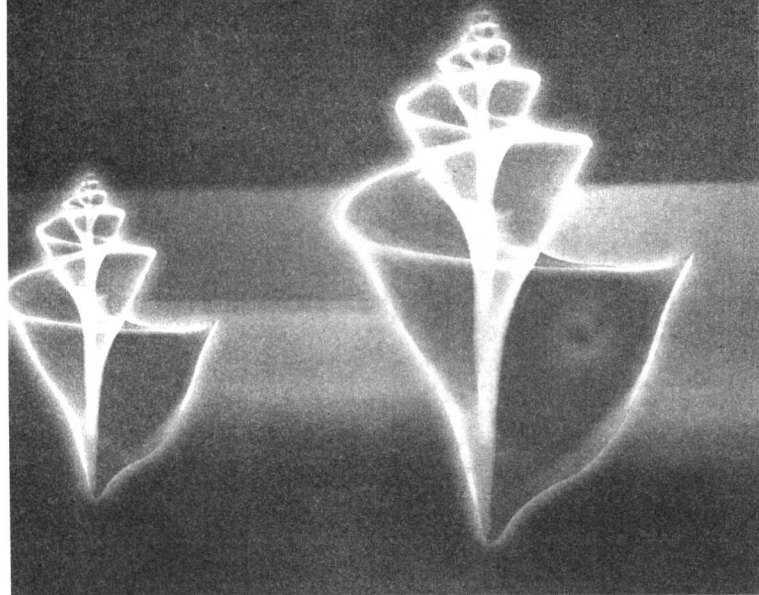


清华大学出版社

二级 C++ 语言程序设计

全国计算机等级考试专业网站 策划

李 季 黄志雄 等编著



全国计算机等级考试 专家讲评

清华大学出版社
北 京

内 容 简 介

本书根据教育部考试中心 2004 年制订的《全国计算机等级考试 考试大纲》编著而成。全书共分 15 章, 前 14 章是二级考试公共基础部分和 C++ 语言程序设计的内容, 每一章中都包含考试大纲和提示、知识点归纳、典型题分析、习题和参考答案; 第 15 章是上机指导, 详细说明了本科目上机题所涉及的知识 and 题型, 并详尽地分析了典型的上机题; 最后提供两套全真笔试模拟试卷并给出答案, 供考生考前实战练习。

本书重点突出、内容丰富、讲解精辟, 适合报考全国计算机等级考试——二级 C++ 语言程序设计的考生考前复习使用。

版权所有, 翻印必究。举报电话: 010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目 (CIP) 数据

二级 C++ 语言程序设计 / 李季等编著. —北京: 清华大学出版社, 2004. 10

(全国计算机等级考试专家讲评)

ISBN 7-302-09251-6

I. 二… II. 李… 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 (2004) 第 084135 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

组稿编辑: 索 梅

文稿编辑: 汪汉友

封面设计: 艺 铭 DESIGN

印 装 者: 北京国马印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 18.25 字数: 429 千字

版 次: 2004 年 10 月第 1 版 2004 年 10 月第 1 次印刷

书 号: ISBN 7-302-09251-6/TP·6492

印 数: 1~5000

定 价: 24.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770175-3103 或 (010)62795704

前 言

全国计算机等级考试 (NCRE) 是面向社会的计算机应用能力水平考试。自 1994 年举办以来, 得到了社会的广泛认可, 是目前国内参加人数最多、影响最大的计算机类考试。截至 2003 年 10 月, 全国计算机等级考试累计考生人数突破 1000 万, 累计获得证书人数将近 300 万。该项考试在促进计算机知识的普及和计算机应用技术的推广、满足用人单位考核工作人员应用计算机的水平等方面适应了社会的需要, 为国家职业技术教育、继续教育作出了较大贡献。

为了不断适应信息技术的发展和社会需求的变化, 教育部考试中心于 2004 年制订了新的等级考试大纲 (2004 年版), 对考试内容和形式进行了一定程度的调整。为了让更多的考生能顺利通过全国计算机等级考试, 帮助考生快速深入地理解考试内容, 我们根据 2004 年版考试大纲的要求, 仔细分析和研究了教育部考试中心的相关资料, 按照考试中心确定的考试范围和考试重点进行编写, 使得本书的每一句话、每一道题都具有针对性和目的性, 让读者能在最短的时间内复习本科目考试所涉及到的所有知识, 做更多的练习题, 从而轻松通过考试。本书的主要内容如下:

(1) 前 4 章是公共基础部分的内容, 第 5 章~第 15 章是有关 C++ 语言程序设计的内容。在每一章中, 基本上都有如下组成。

① 考试大纲和提示。每章前面明确列出教育部考试中心的等级考试大纲; 考试大纲提示中, 分析说明本章内容在考试中可能的出题方向、题型以及重点和难点。

② 知识点归纳。知识点归纳的内容简明扼要地列出本科考试的知识点, 有利于读者全面快速地复习和回顾重要的知识。

③ 典型题分析。本部分的目的是让考生接触各类典型考题, 并熟悉每一类题的出题陷阱和思考方法。

④ 习题和参考答案。要考得高分, 必须多动手训练, 这是看似简单却十分重要的经验。习题和参考答案在本书中占有不少的篇幅。本书的例题和习题都按照真实考试的题型出现, 针对笔试中的选择、填空两种题型, 全书按章节顺序介绍不同类型试题的解题思路和方法, 以便考生了解各章的重点和难点, 提高解题速度, 掌握解题技巧。

(2) 第 15 章是上机指导部分, 详细说明了本科目上机题所涉及的知识、算法、题型, 并详尽地分析了典型的上机题。读者如果需要本上机指导部分的习题, 可以从清华大学出版社的网站下载 (网址为: <http://www.tup.com.cn>), 或通过电子邮件向我们索取 (xiaoxiang-007@sohu.com)。

(3) 全真笔试模拟试题。我们精心设计了两套笔试的全真模拟题, 并给出答案, 力图从深度和广度上体现出计算机等级考试中的难度和水平, 读者可以进行实战演练, 看看自己的水平如何。

目 录

第1章 数据结构与算法	1
考纲	1
考纲提示	1
知识点归纳	1
【算法的基本概念】	1
【算法的复杂度】	2
【数据结构】	2
【数据的逻辑结构】	2
【数据的存储结构】	3
【数据的运算】	3
【数据结构的图形表示】	3
【线性结构与非线性结构】	3
【线性表】	4
【线性表的顺序存储结构】	4
【线性表的插入运算】	5
【线性表的删除运算】	5
【栈】	5
【栈的顺序存储结构】	6
【栈的基本运算】	6
【队列】	6
【队列的顺序存储结构】	7
【队列的基本运算】	7
【线性单链表】	7
【循环链表】	7
【双向链表】	7
【链表的基本操作】	8
【树及其基本概念】	8
【二叉树】	9
【二叉树的存储结构】	9
【二叉树的遍历】	11
【查找】	12
【排序】	12
典型题分析	13
习题	17

参考答案	19
第 2 章 程序设计基础	20
考纲	20
考纲提示	20
知识点归纳	20
【程序设计的方法】	20
【程序设计的风格】	20
【结构化程序设计的主要原则】	22
【结构化程序的基本结构与特点】	22
【面向对象的程序设计】	23
【面向对象方法的基本概念】	23
典型题分析	24
习题	26
参考答案	28
第 3 章 软件工程基础	29
考纲	29
考纲提示	29
知识点归纳	29
【软件定义与特点】	29
【软件危机与软件工程】	29
【软件工程过程】	30
【软件生命周期】	30
【软件工程的目标与原则】	30
【软件工具与软件开发环境】	31
【结构化分析方法】	31
【数据流图】	32
【数据字典】	32
【软件需求】	33
【结构化设计】	33
【总体设计】	33
【详细设计】	34
【软件测试的目的】	34
【软件测试的准则】	34
【软件测试技术与方法综述】	34
【软件测试的实施】	36
【程序调试】	37

【常见调试方法】	37
典型题分析	37
习题	40
参考答案	41
第 4 章 数据库设计基础	42
考纲	42
考纲提示	42
知识点归纳	42
【数据库】	42
【数据库管理系统】	42
【数据库系统】	43
【数据库系统的基本特点】	43
【数据库系统的内部结构体系】	43
【数据模型】	43
【实体联系模型】	44
【E-R 图】	44
【关系数据模型】	46
【E-R 模型到关系模型的转换】	46
【关系代数运算】	46
【数据库的规范化】	47
【数据库设计】	48
【需求分析】	48
【概念设计】	48
【逻辑设计】	49
【物理设计】	49
典型题分析	50
习题	53
参考答案	55
第 5 章 C++ 语言概述	56
考纲	56
考纲提示	56
知识点归纳	56
【C++ 语言的发展及其特点】	56
【面向对象语言的要素】	56
【C++ 对面向对象程序设计方法的支持】	57
【C++ 语言的词法和词法规则】	57

【C++程序结构】	59
【C++程序的编辑、编译、运行与调试】	60
典型题分析	61
习题	64
参考答案	66
第 6 章 C++数据类型	67
考纲	67
考纲提示	67
知识点归纳	67
【基本数据类型】	67
【导出类型】	68
【常量】	69
【变量】	70
【运算符】	71
【算术运算中的类型转换】	73
【表达式】	74
【复合数据类型】	74
典型题分析	75
习题	82
参考答案	83
第 7 章 控制语句	84
考纲	84
考纲提示	84
知识点归纳	84
【C++语言的语句】	84
【顺序控制语句】	86
【选择控制语句】	86
【循环控制语句】	87
【跳转语句】	88
【循环嵌套语句】	89
典型题分析	89
习题	96
参考答案	100
第 8 章 数组、指针与引用	101
考纲	101
考纲提示	101

知识点归纳	101
【数组的基本概念】	101
【字符串与字符数组】	102
【指针】	104
【指针和数组之间的关系】	106
【引用】	106
典型题分析	107
习题	121
参考答案	125
第9章 函数	126
考纲	126
考纲提示	126
知识点归纳	126
【函数概述】	126
【函数的定义和调用】	127
【函数的类型和返回值】	128
【函数的参数】	128
【带有默认参数值的函数】	129
【标识符的作用域】	129
【全局变量与局部变量】	130
【变量的存储类别】	130
【递归函数】	131
【内联函数】	132
【函数重载】	132
【预处理】	132
典型题分析	133
习题	144
参考答案	149
第10章 类和对象	150
考纲	150
考纲提示	150
知识点归纳	150
【类】	150
【对象】	152
【构造函数与析构函数】	153
【对象的生存周期】	154

【静态成员】	154
【常类型】	155
【this 指针】	156
【友元】	156
【对象数组与成员对象】	157
典型题分析	158
习题	172
参考答案	176
第 11 章 派生与继承	177
考纲	177
考纲提示	177
知识点归纳	177
【派生类】	177
【继承基类的成员】	178
【指针的使用】	179
【虚基类】	180
典型题分析	181
习题	190
参考答案	196
第 12 章 多态性和虚函数	197
考纲	197
考纲提示	197
知识点归纳	197
【多态性的概念】	197
【虚函数机制的要点】	197
【虚函数】	198
【纯虚函数和抽象类】	199
【运算符重载】	199
典型题分析	201
习题	207
参考答案	210
第 13 章 模板	211
考纲	211
考纲提示	211
知识点归纳	211
【函数模板】	211

【类模板】	212
典型题分析	213
习题	216
参考答案	219
第 14 章 C++流	220
考纲	220
考纲提示	220
知识点归纳	220
【C++流】	220
【I/O 格式的控制】	222
【C++文件流】	224
典型题分析	227
习题	231
参考答案	235
第 15 章 C++上机指导	236
考纲	236
考纲提示	236
知识点归纳	236
【上机考试时间和题型】	236
【使用 Visual C++ 6.0 编写和调试 C++程序】	236
【常用算法】	237
【上机考试重点内容】	242
典型题分析	243
习题	256
参考答案	260
全国计算机等级考试二级 C++语言程序设计全真笔试模拟试卷（一）	262
参考答案	269
全国计算机等级考试二级 C++语言程序设计全真笔试模拟试卷（二）	270
参考答案	278

第 1 章 数据结构与算法

考纲

1. 算法的基本概念; 算法复杂度的概念和意义。
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)。

考纲提示

本章主要考查数据结构及相关基本概念、几种典型的数据结构及其操作、算法的概念及算法复杂度、主要的查找及排序算法。在新考试大纲的公共基础部分中, 约占 30% 的比例。

知识点归纳

【算法的基本概念】

所谓算法是指解题方案的准确而完整的描述。它是对特定问题求解步骤的一种描述, 它是指令的有限序列, 其中每条指令表示一个或是多个操作。严格说来, 一个算法必须具有下列 5 个主要特性。

(1) 有穷性。一个算法必须总是 (对任何合法的输入值) 在执行有穷步之后结束, 而且每一步都可在有穷时间内完成。

(2) 确定性。算法中每条指令必须有确切含义, 且在任何条件下, 算法只有惟一的一条执行路径。

(3) 可行性。算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现的。

(4) 有输入。一个算法有 0 个或多个输入, 这些输入取自于某个特定的对象集合。

(5) 有输出。一个算法有 0 个或多个输出, 这些输出是同输入有着某些特定关系的量。

综上所述, 所谓算法, 是一组严谨地定义运算顺序的规则, 而且每一个规则都是有效

且明确的, 此顺序将在有限的次数下终止。

【算法的复杂度】

算法的复杂度是本章的重点也是难点。

选用算法首先考虑正确性, 还要考虑执行算法所耗费的时间和存储空间, 同时, 算法应易于理解、编码、调试等。算法的复杂度可分为时间复杂度和空间复杂度, 是衡量算法优劣的量度。

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。一般情况下, 算法中的基本操作重复执行的次数是问题规模 n 的某个函数 $f(n)$, 算法的时间量度记作: 算法的工作量 $= f(n)$ 。它表示随问题规模 n 的增大, 算法执行时间的增长率和 $f(n)$ 的增长率相同, 称作算法的渐进时间复杂度, 简称时间复杂度。

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。类似时间复杂度, 空间复杂度作为算法所需存储空间的量度: $S(n) = O(g(n))$ 。

其中, n 为问题的规模, 表示随着问题规模 n 的增大, 算法运行所需存储量的增长率与 $g(n)$ 的增长率相同。

【数据结构】

利用计算机进行数据处理是计算机应用的一个重要领域。数据结构作为计算机的一门学科, 主要研究和讨论以下 3 方面的问题:

- (1) 数据集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。

简单地说, 数据结构就是数据的数据模型。一般说来, 用计算机解决一个具体问题时, 大致需要经历下列几个步骤:

- (1) 首先从具体问题抽象出一个适当的数学模型;
- (2) 然后设计一个解此数学模型的算法;
- (3) 最后编出程序、进行测试、调整, 直至得到最终解答。

寻求数学模型的实质是分析问题, 从中提取操作的对象, 并找出这些操作对象之间含有的关系, 然后用数学的语言加以描述。

【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之间的前后件关系。因此, 所谓结构, 实际上就是指数据元素之间的前后件关系。

数据的逻辑结构是指数据元素之间的逻辑关系, 它可以用一个数据元素的集合和定义在此集合上的若干关系来表示。

数据的逻辑结构是从逻辑关系上描述数据, 它与数据在计算机中的存储位置无关, 是独立于计算机的。

【数据的存储结构】

数据的存储结构是本章的重点知识点。它是数据元素及其关系在计算机存储器内的表示。数据的存储结构是逻辑结构用计算机语言的实现，即建立数据的机内表示。存储结构的主要内容是指在存储空间中使用一个存储结点来存储一个数据元素；在存储空间中建立各存储结点之间的关联，以表示数据元素之间的逻辑关系。其中，存储结点是指一个数据元素在存储结构中的存储。

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，常用数据的存储结构有如下的4种。

(1) 顺序存储方式。每一个存储结点只含一个数据元素。所有的存储结点相继存储在一个连续的存储区里。用存储结点之间的位置关系表示数据元素之间的逻辑关系。

(2) 链式存储方式。每一个存储结点不仅含有各数据元素，还包括指针。每一个指针指向一个与本结点有逻辑关系的结点，即用指针表示逻辑关系。

(3) 索引存储方式。每一个存储结点仅含一个数据元素，所有的存储结点都连续存放。此外，增设一个索引表。

(4) 散列存储方式。每一个存储结点仅含一个数据元素，数据元素按散列函数确定存储位置。

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。

【数据的运算】

数据运算即对数据施加的操作。常用的运算如下。

- (1) 查找运算。从结构中找出满足某种条件的结点的位置。
- (2) 读取运算。读出结构中指定位置上的内容。
- (3) 插入运算。在结构中的某指定位置上增加一个新的结点。
- (4) 删除运算。撤销结构中指定位置上的结点。
- (5) 更新运算。修改结构中某指定结点的内容。

【数据结构的图形表示】

一个数据结构除了用二元关系表示外，还可以直接用图形表示。图 1.1 是一些常见数据结构的图形表示示例。

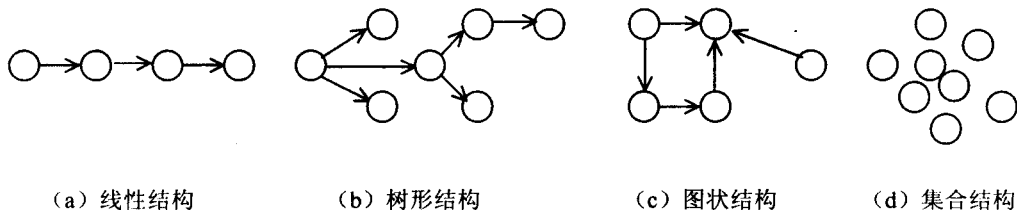


图 1.1 常见数据结构的图形表示示例

【线性结构与非线性结构】

根据数据结构中各数据元素之间前后件关系的复杂程度，将数据结构分成两大类：线

性结构和非线性结构。

1. 线性结构

在数据元素的非空有限集中，线性结构的逻辑特征如下：

- (1) 存在惟一的一个被称作“第一个”的数据元素；
- (2) 存在惟一的一个被称作“最后一个”的数据元素；
- (3) 除第一个之外，集合中的每个数据元素均只有一个前驱；
- (4) 除最后一个之外，集合中的每个数据元素均只有一个后继。

线性表是一个线性结构。

2. 非线性结构

非线性结构的逻辑特征是：一个结点可能有多个直接前驱和直接后继。例如，树和图都是非线性结构。

线性结构与非线性结构都可以是空的数据结构。一个空的数据结构究竟属于线性结构还是非线性结构，要根据具体情况来确定。如果对该数据结构的运算是按线性结构的规则来处理的，则属于线性结构；否则属于非线性结构。

【线性表】

线性表是最简单、最常用的一种数据结构。所谓线性表是 n 个数据元素的有限序列。至于每个数据元素的具体含义在不同情况下各不相同，它可以是一个数，或是一个符号，也可以是一页书，甚至其他更复杂的信息。在不同的情况下，它可以有不同的含义。

例如，26 个英文字母的字母表 (A, B, C, D, ..., Z) 是一个线性表，表中的数据元素是单个字母。

又如，某校 1998—2004 年的计算机拥有量的变化情况，可以用线性表的形式给出：(23, 35, 67, 156, 240, 287, 324)。表中的数据元素是整数。

【线性表的顺序存储结构】

线性表的顺序存储结构指的是用一组地址连续的存储单元依次存储线性表中的数据元素。其特点如下：

- (1) 线性表中所有元素所占的存储空间是连续的；
- (2) 线性表中各数据元素在存储空间中是按逻辑顺序依次存放的。

可以看出，在线性表的顺序存储结构中，其前后件两个元素在存储空间中是紧邻的，且前件元素一定存储在后件元素的前面。

图 1.2 说明了数据元素在计算机内的存储情况。其中 $a_1, a_2, \dots, a_n$ 表示线性表中的数据元素。

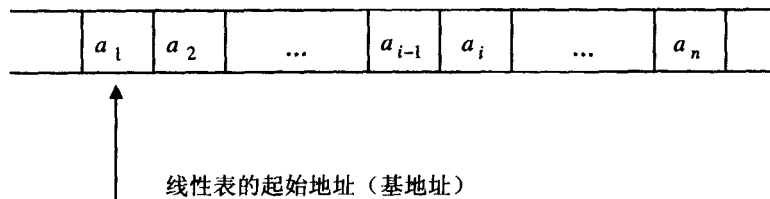


图 1.2 线性表的顺序存储结构示意图

在线性表的顺序存储结构中, 如果线性表中各数据元素所占的存储空间(字节数)相等, 则要在该线性表中查找某一个元素是很方便的。

所有数据元素的存储位置均取决于第一个数据元素的存储位置, 即:

$$\text{LOC}(a_i) = \text{LOC}(a_1) + (i-1) \times C$$

$\uparrow$
基地址

$\uparrow$
一个数据元素所占字节数

【线性表的插入运算】

插入和删除运算是线性表的基本操作。所谓插入运算是指在结构中的某指定位置上增加一个新结点; 而删除运算是指撤销结构中某结点的内容。下面依次进行详细讨论。

线性表的插入操作是指在线性表的第 $i-1$ 个数据元素和第 i 个数据元素之间插入一个新的数据元素, 就是要使长度为 n 的线性表

$$(a_1, \dots, a_{i-1}, a_i, \dots, a_n)$$

变成长度为 $n+1$ 的线性表

$$(a_1, \dots, a_{i-1}, b, a_i, \dots, a_n)$$

数据元素 a_{i-1} 和 a_i 之间的逻辑关系发生了变化。

一般情况下, 要在第 i ($1 \leq i \leq n$) 个元素之前插入一个新元素时, 首先要从最后一个(即第 n 个)元素开始, 直到第 i 个元素之间共 $n-i+1$ 个元素依次向后移动一个位置, 移动结束后, 第 i 个位置就被空出, 然后将新元素插入到第 i 项。插入结束后, 线性表的长度就增加了 1。

平均情况下, 要在线性表中插入一个新元素, 需要移动表中一半的元素。因此, 在线性表顺序存储的情况下, 要插入一个新元素, 其效率是很低的, 特别是在线性表比较大的情况下更为突出, 由于数据元素的移动而消耗较多的处理时间。

【线性表的删除运算】

相反, 线性表的删除操作是使长度为 n 的线性表

$$(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$$

变为长度为 $n-1$ 的线性表

$$(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$$

数据元素 a_{i-1} , a_i 和 a_{i+1} 之间的逻辑关系发生了变化。

一般情况下, 要删除第 i ($1 \leq i \leq n$) 个元素时, 要从第 $i+1$ 个元素开始, 直到第 n 个元素之间共 $n-i$ 个元素依次向前移动一个位置。删除结束后, 线性表的长度就减小了 1。

平均情况下, 要在线性表中删除一个元素, 需要移动表中一半的元素。因此, 在线性表顺序存储的情况下, 要删除一个元素, 其效率也是很低的, 特别是在线性表比较大的情况下更为突出, 由于数据元素的移动而消耗较多的处理时间。

【栈】

栈是限定仅在表尾进行插入和删除操作的线性表。因此, 对栈来说, 表尾端有其特殊的含义, 称为栈顶, 相应地, 表头端称为栈底。栈顶元素总是最后被插入的元素, 从而也是最先能被删除的元素; 栈底元素总是最先被插入底元素, 从而也是最后才能被删除底元

素。栈的修改是按“后进先出”或“先进后出”的原则进行的。因此，栈又称先进后出表或后进先出表。

【栈的顺序存储结构】

栈的顺序存储结构是利用一组地址连续的存储单元依次存放自栈底到栈顶的数据元素，同时附设指针指示栈顶元素在顺序栈中的位置，如图 1.3 所示。

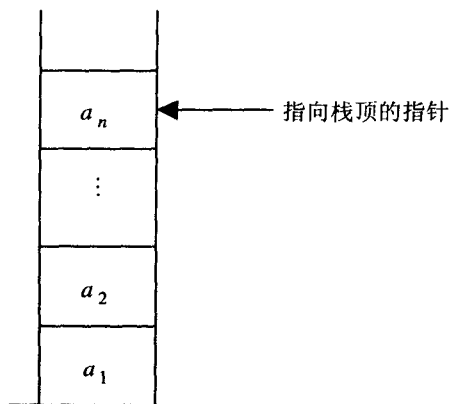


图 1.3 栈的顺序存储结构示意图

在图 1.3 中， a_1 为栈底元素， a_n 为栈顶元素。栈中的元素按照 $a_1, a_2, \dots, a_n$ 的次序进栈，退栈的第一个元素为栈顶元素 a_n 。

【栈的基本运算】

栈的基本运算有 3 种：入栈、出栈与读栈顶元素。

1. 入栈运算

入栈运算是指在栈顶插入一个新元素。可分为两个基本操作：首先将栈顶指针进一，然后将新元素插入到栈顶指针指向的位置。

当栈顶指针已经指向存储空间的最后一个位置时，说明栈空间已满，不能再进行入栈操作。

2. 退栈运算

退栈运算是指取出栈顶元素并赋给一个指定的变量。可分为两个基本操作：首先将栈顶元素（栈顶指针指向的元素）赋给一个指定的变量，然后将栈顶指针退一。

当栈顶指针为 0 时，说明栈空，不可能进行退栈操作。

3. 读栈顶元素

读栈顶元素是指将栈顶元素赋给一个指定的变量。必须注意，这个运算不删除栈顶元素，只是将它的值赋给一个变量，因此，在这个运算中，栈顶指针不会改变。

【队列】

队列是限定仅在表的一端进行插入，而在表的另一端删除数据元素的线性表。在队列中，允许插入的一端叫做队尾，允许删除的一端则叫做队头。