

National Computer Rank Examination

全国计算机等级考试

专用辅导教程:

二级 Access

希赛教育等考学院

朱卿 桂阳 主编



- ◆紧扣最新考试大纲，透彻精讲大纲规定考点
- ◆突出重点与难点，深入分析例题，讲练结合
- ◆提供最新真题解析，摸清考试规律，掌握实考难度



光盘特色

- ◆配备3种模拟方式（固定抽题、随机抽题、重复抽题）和大量题库，模拟实考环境，反复练习，力保顺利通过考试
- ◆灵活的模拟操作界面，随时自测，实时查看成绩，做到心中有数
- ◆盘内免费提供精选的教学视频，名师讲解，学习更轻松
- ◆丰富的网络配套资源，涵盖论坛答疑、网络课堂、在线测试



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

全国计算机等级考试专用辅导丛书
National Computer Rank Examination

National Computer Rank Examination

全国计算机等级考试

专用辅导教程：

二级 Access

图书馆

希赛教育等考学院

朱卿 桂阳 主编

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书由希赛教育等考学院组织编写,作为全国计算机等级考试二级Access数据库的辅导和培训指定教程。书中内容紧扣教育部考试中心新推出的考试大纲,通过对历年试题进行科学分析、研究、总结、提炼而成。书中内容全面实用,涵盖了考试大纲规定的所有知识点,对考试大纲规定的内容有重点地进行了细化和深化。阅读本书,就相当于阅读了一本详细的、带有知识注释的考试大纲。准备考试的人员可通过阅读本书掌握考试大纲规定的知识,掌握考试重点和难点,熟悉内容的分布。

本书适合参加全国计算机等级考试的人员及广大计算机爱好者阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

全国计算机等级考试专用辅导教程:二级 Access /朱卿,桂阳主编. — 北京:电子工业出版社,2011.1
(全国计算机等级考试专用辅导丛书)

ISBN 978-7-121-12000-8

I. ①全… II. ①朱… ②桂… III. ①关系数据库—数据库管理系统, Access—水平考试—自学参考资料
IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2010)第 199113 号

责任编辑:李云静

印 刷:北京中新伟业印刷有限公司

装 订:北京中新伟业印刷有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:860×1092 1/16 印张:14.5 字数:506 千字

印 次:2011 年 1 月第 1 次印刷

印 数:4000 册 定价:33.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

全国计算机等级考试(NCRE)由教育部考试中心主办,面向社会,用于考查非计算机专业人员计算机应用知识与能力。考试客观、公正,得到了社会的广泛认可。

本书根据全国计算机等级考试二级 Access 数据库的最新考试大纲编写而成,在组织和写作上倾注了作者们的许多精力和心血,相信能够对考生提高通过率,有效地完成“考试过关”提供帮助。考生可通过阅读本书,迅速掌握考试所涉及的知识点,全面梳理和系统学习考试大纲中的内容。

作者权威,阵容强大

希赛教育(www.educity.cn)专业从事人才培养、教育产品开发、教育图书出版,在职业教育方面具有极高的权威性。特别是在在线教育方面名列前茅,希赛教育的远程教育模式得到了国家教育部门的认可和推广。

希赛教育等考学院是国内首屈一指的进行计算机等级考试在线教育的大型教育机构,在该领域取得了很好的效果。我们组织大纲制订者和阅卷组成员编写了考试辅导教材近 20 本,内容涵盖了计算机等级考试的主要级别。组织权威专家和辅导名师录制了考试培训视频教程,对历年考试进行了跟踪研究和比较研究,编写了权威的全真模拟试题。希赛教育的计算机等级考试培训采取统一教材、统一视频、统一认证教师的形式,采取线下培训与线上辅导相结合的方式,确保学员在通过考试的前提下能真正学到有用的知识。

本书由希赛教育等考学院组织编写,参加编写的人员来自大学教学一线和企业研发团队,具有丰富的教学和辅导经验,对等级考试有深入的研究,具有极强的应试技巧、理论知识、实践经验和责任心。

本书由朱卿和桂阳主编,张友生审核了所有稿件。参加编写工作的有张友生、朱卿、周玲、桂阳、胡钊源、刘毅、施游、朱小平、符春、王勇、唐小娟。

在线测试,心中有数

上学吧在线测试平台(www.shangxueba.com)为考生准备了在线测试,其中有数十套全真模拟试题和考前密卷,考生可选择任何一套进行测试。测试完毕,系统自动判卷,立即给出分数。

对于考生做错的地方,系统会自动记忆,待考生第二次参加测试时,可选择“试题复习”。这样,系统就会自动把考生原来做错的试题显示出来,供考生重新测试,以加强记忆。

因此,读者可利用上学吧在线测试平台的在线测试系统检查自己的实际水平,加强考前训练,做到心中有数,考试不慌。

诸多帮助,诚挚致谢

在本书出版之际,要特别感谢教育部考试中心计算机等级考试办公室的命题专家们,编者在本书中引用了部分考试原题,使本书能够尽量方便读者的阅读。在本书的编写过程中,参考了许多相关的文献和书籍,编者在此对这些参考文献的作者表示感谢。

感谢电子工业出版社田小康老师,他在本书的策划、选题的申报、写作大纲的确定,以及编辑、出版等方面,付出了辛勤的劳动和智慧,给予了我们很多的支持和帮助。

感谢参加希赛教育计算机等级考试辅导和培训的学员,正是他们的想法汇成了本书的源动力,他们的意见使本书更加贴近读者。

由于编者水平有限,且本书涉及的内容很广,书中难免存在错漏和不妥之处,编者诚恳地期望各位专家和读者不吝指正和帮助,对此,我们将十分感激。

互动讨论,专家答疑

希赛教育等考学院(www.csaidk.com)是中国领先的计算机等级考试在线教育网站,该网站论坛是国内人气很旺的计算机等级考试社区,在这里,读者可以和数百万考生进行在线交流,讨论有关学习和考试的问题,以及人生和职业规划的话题。希赛教育等考学院拥有强大的师资队伍,为读者提供全程的答疑服务,在线回答读者的提问。

有关本书的意见反馈和咨询,读者可在希赛教育等考学院论坛“等级考试教材”板块中的“希赛教育等考学院”栏目中与作者进行交流。

希赛教育等考学院

目 录

第 1 章 算法和数据结构..... 1	第 2 章 程序设计结构..... 17
1.1 算法与数据结构概述..... 1	2.1 程序设计方法与风格..... 17
1.1.1 算法的概念..... 1	2.2 结构化程序设计..... 18
1.1.2 算法的复杂度..... 2	2.3 面向对象的程序设计..... 18
1.1.3 数据结构的定义..... 3	2.3.1 面向对象的特点..... 19
1.1.4 数据结构的表示..... 3	2.3.2 类和实例..... 19
1.1.5 线性结构与非线性结构..... 4	2.3.3 消息..... 19
1.2 线性表..... 4	2.4 习题..... 20
1.2.1 线性表概述..... 4	2.4.1 选择题..... 20
1.2.2 线性表的顺序存储..... 4	2.4.2 填空题..... 20
1.3 栈和队列..... 5	第 3 章 软件工程基础..... 21
1.3.1 栈的定义与操作..... 6	3.1 软件工程的基本概念..... 21
1.3.2 队列的定义与操作..... 6	3.1.1 软件的含义..... 21
1.4 线性链表..... 7	3.1.2 软件工程..... 22
1.4.1 线性表的链式存储..... 7	3.2 结构化分析方法..... 23
1.4.2 双向链表的结构及其基本运算..... 8	3.2.1 结构化分析方法概述..... 23
1.5 树与二叉树..... 8	3.2.2 软件需求规格说明书..... 24
1.5.1 树的定义..... 9	3.3 结构化设计方法..... 25
1.5.2 二叉树的定义及其性质..... 9	3.3.1 软件设计的基本内容..... 25
1.5.3 二叉树的遍历..... 10	3.3.2 结构化设计..... 26
1.6 查找技术..... 11	3.3.3 概要设计..... 27
1.6.1 顺序查找..... 11	3.3.4 详细设计..... 28
1.6.2 二分法查找..... 12	3.4 软件测试..... 28
1.7 排序技术..... 12	3.4.1 软件测试概述..... 28
1.8 习题..... 14	3.4.2 软件测试技术..... 29
1.8.1 选择题..... 14	3.5 程序的调试..... 30
1.8.2 填空题..... 16	

3.5.1 步骤与方法	30	5.3.2 数据库设计的步骤	51
3.5.2 静态调试	31	5.4 Access 系统简介	52
3.5.3 动态调试	32	5.4.1 Access 的特点与功能	52
3.6 习题	32	5.4.2 Access 系统结构	52
3.6.1 选择题	32	5.4.3 SQL 语言简介	53
3.6.2 填空题	33	5.4.4 Access 基本操作	53
第 4 章 数据库设计基础	34	5.5 本章习题	55
4.1 数据库的基本概念	34	5.5.1 选择题	55
4.1.1 数据和信息	34	5.5.2 填空题	57
4.1.2 数据处理、数据库与数据库管理系统	34	第 6 章 数据库和表	58
4.1.3 数据库系统的发展	35	6.1 创建数据库	58
4.1.4 数据库系统的内部结构体系	36	6.1.1 创建空数据库	58
4.2 数据模型	37	6.1.2 使用向导创建数据库	59
4.2.1 数据模型概述	37	6.1.3 数据库简单操作	60
4.2.2 E-R 模型	38	6.2 建立表结构	60
4.2.3 关系模型	39	6.2.1 Access 数据类型和字段属性	60
4.2.4 数据操作	40	6.2.2 建立表结构的方式	63
4.2.5 关系中的数据约束	40	6.2.3 建立表之间的关系	64
4.3 关系代数	41	6.2.4 向数据表添加数据	65
4.4 数据库设计	41	6.3 维护表	66
4.5 习题	42	6.3.1 修改表结构	67
4.5.1 选择题	42	6.3.2 编辑表内容	68
4.5.2 填空题	43	6.3.3 调整表的外观	68
第 5 章 数据库基础知识	44	6.4 操作表	70
5.1 数据库基础知识概述	44	6.4.1 查找数据	70
5.1.1 计算机数据管理发展阶段	44	6.4.2 替换数据	71
5.1.2 数据库管理系统	46	6.4.3 排序记录	71
5.1.3 数据模型	47	6.4.4 筛选记录	72
5.2 关系数据库	49	6.5 本章习题	73
5.2.1 关系数据库的基本概念	49	6.5.1 选择题	73
5.2.2 关系运算	50	6.5.2 填空题	75
5.3 数据库设计基础	51	第 7 章 数据库查询	76
5.3.1 数据库设计的原则	51	7.1 查询概述	76

7.1.1 查询的功能.....	76	8.2 创建窗体.....	95
7.1.2 查询的分类.....	77	8.2.1 用“窗体向导”创建窗体.....	95
7.1.3 查询条件.....	78	8.2.2 创建图表窗体.....	96
7.2 创建选择查询.....	79	8.3 设计窗体.....	96
7.2.1 使用向导创建查询.....	79	8.3.1 窗体设计视图.....	96
7.2.2 使用设计器创建查询.....	80	8.3.2 窗体控件介绍.....	97
7.2.3 在查询中计算.....	80	8.3.3 窗体控件属性与事件.....	99
7.3 创建交叉表查询.....	81	8.4 窗体控件格式化.....	100
7.3.1 使用交叉表查询向导.....	82	8.4.1 窗体格式设定.....	100
7.3.2 使用设计视图创建交叉查询.....	82	8.4.2 添加日期和时间.....	101
7.4 创建参数查询.....	82	8.4.3 对齐窗体控件.....	101
7.4.1 单参数查询.....	83	8.5 本章习题.....	102
7.4.2 多参数查询.....	83	8.5.1 选择题.....	102
7.5 创建操作查询.....	83	8.5.2 填空题.....	103
7.5.1 生成表查询.....	83	第9章 报表.....	104
7.5.2 删除查询.....	84	9.1 报表概述.....	104
7.5.3 更新查询.....	84	9.1.1 报表的基本概念.....	104
7.5.4 追加查询.....	84	9.1.2 报表设计区.....	105
7.6 创建 SQL 查询.....	84	9.1.3 报表的分类.....	105
7.6.1 SQL 语言.....	84	9.2 创建报表.....	106
7.6.2 创建 SQL 查询.....	85	9.2.1 使用自动创建报表创建.....	106
7.6.3 创建 SQL 特定查询.....	88	9.2.2 使用报表向导创建.....	106
7.7 操作已创建的查询.....	88	9.2.3 使用图表向导创建.....	106
7.7.1 运行查询.....	88	9.2.4 使用标签向导创建.....	107
7.7.2 编辑生成的查询.....	88	9.2.5 使用设计视图创建.....	107
7.7.3 查询结果排序.....	89	9.3 编辑报表.....	108
7.8 本章习题.....	90	9.3.1 编辑报表格式.....	108
7.8.1 选择题.....	90	9.3.2 添加报表元素.....	108
7.8.2 填空题.....	92	9.4 排序和分组.....	110
第8章 窗体.....	93	9.4.1 数据记录排序.....	110
8.1 窗体概述.....	93	9.4.2 数据记录分组.....	110
8.1.1 窗体的类型.....	94	9.5 报表控件应用.....	110
8.1.2 窗体视图.....	94	9.5.1 添加报表计算控件.....	110

9.5.2 报表统计计算	110	11.1.1 宏的概念与功能	120
9.5.3 报表常用函数	111	11.1.2 设置宏操作	121
9.6 创建子报表	111	11.2 创建宏	121
9.6.1 子报表概述	111	11.2.1 创建操作宏	122
9.6.2 子报表创建与添加	111	11.2.2 创建宏组	122
9.6.3 创建多列报表	112	11.2.3 创建条件宏	122
9.7 复杂报表设计	112	11.2.4 设置宏参数	123
9.7.1 报表属性	112	11.2.5 运行宏	123
9.7.2 节属性	112	11.2.6 调试宏	123
9.8 本章习题	113	11.3 宏的事件触发	124
9.8.1 选择题	113	11.3.1 事件的概念	124
9.8.2 填空题	114	11.3.2 通过事件触发宏	124
第 10 章 数据访问页	115	11.4 本章习题	124
10.1 数据访问页概述	115	11.4.1 选择题	124
10.1.1 数据访问页的视图	116	11.4.2 填空题	126
10.1.2 数据访问页的数据来源	116	第 12 章 模块与 VBA 编程	127
10.2 创建数据访问页	116	12.1 模块的概述	127
10.2.1 自动创建访问页	116	12.1.1 类模块	127
10.2.2 使用向导创建访问页	117	12.1.2 标准模块	128
10.2.3 使用设计视图创建访问页	117	12.1.3 将宏转换为模块	128
10.3 编辑数据访问页	117	12.2 创建模块	128
10.3.1 添加标签	117	12.2.1 在模块中加入过程	128
10.3.2 添加按钮	117	12.2.2 在模块中执行宏	129
10.3.3 添加滚动文字	117	12.3 VBA 程序设计概述	129
10.3.4 设置背景	118	12.3.1 面向对象的程序设计	129
10.4 数据访问页应用	119	12.3.2 VBA 编程环境	130
10.4.1 浏览数据访问页	119	12.3.3 数据类型与数据库对象	131
10.4.2 删除数据访问页	119	12.3.4 变量与常量	132
10.5 本章习题	119	12.3.5 过程与函数	135
10.5.1 选择题	119	12.4 VBA 流程控制	136
10.5.2 填空题	119	12.4.1 赋值语句	137
第 11 章 宏	120	12.4.2 条件语句	137
11.1 宏的概述	120	12.4.3 循环语句	139

12.5 过程调用和参数传递.....	142	13.4 上机模拟试题三.....	164
12.5.1 过程调用.....	142	13.5 上机模拟试题四.....	165
12.5.2 参数传递.....	143	13.6 上机模拟试题五.....	166
12.6 VBA 错误处理和调试.....	144	13.7 上机模拟试题一参考答案与解析.....	167
12.6.1 错误处理.....	144	13.8 上机模拟试题二参考答案与解析.....	172
12.6.2 VBA 程序调试.....	144	13.9 上机模拟试题三参考答案与解析.....	175
12.7 本章习题.....	145	13.10 上机模拟试题四参考答案与解析.....	177
12.7.1 选择题.....	145	13.11 上机模拟试题五参考答案与解析.....	180
12.7.2 填空题.....	154	附录 A 习题分析与解答.....	183
第 13 章 上机模拟试题与解析.....	161	附录 B 2010 年 4 月笔试试卷.....	214
13.1 上机应试技巧.....	161	附录 C 2010 年 4 月笔试试卷答案及详解.....	219
13.2 上机模拟试题一.....	162		
13.3 上机模拟试题二.....	163		

第1章 算法和数据结构

本章主要介绍算法、线性表、栈和队列、二叉树的概念，介绍几种常见的排序技术。结合计算机等级考试的要求，具体如表 1-1 所示。

表 1-1 考试要求

考试知识点	重要性
算法，线性表基本概念	★
栈和队列	★★★
树和二叉树	★★★★★
查找技术	★
排序技术	★★★★★

1.1 算法与数据结构概述

本节的主要考点集中在算法与数据结构的基本概念上，包括算法的基本特征、复杂度，以及数据结构的表示等。

1.1.1 算法的概念

算法（Algorithm）是一系列解决问题的清晰指令，也就是说，能够对一定规范的输入，在有限时间内获得所要求的输出。如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。

1. 算法的基本特征

- （1）有穷性。一个算法必须保证执行有限步骤之后结束。
- （2）确定性。算法的每一步骤必须有确切的定义。
- （3）可行性。算法原则上能够精确地运行，而且人们用笔和纸做有限次运算后即可完成。

2. 算法的基本要素

（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。

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。在一般的计算机系统中，基本的运算和操作有以下 4 类：

- ①算术运算：主要包括加、减、乘、除等运算。
- ②逻辑运算：主要包括“与”、“或”、“非”等运算。
- ③关系运算：主要包括“大于”、“小于”、“等于”、“不等于”等运算。
- ④数据传输：主要包括赋值、输入、输出等操作。

(2) 算法的控制结构：一个算法的功能不仅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。

3. 算法设计的基本方法

计算机算法不同于人工处理的方法，下面是工程上常用的几种算法设计，在实际应用时，各种方法之间往往存在着一定的联系。

(1) 递推法。递推法是利用问题本身所具有的一种递推关系求问题解的一种方法。它把问题分成若干步，找出相邻几步的关系，从而达到目的，此方法称为递推法。

(2) 递归。递归指的是一个过程：函数不断引用自身，直到引用的对象已知。

(3) 穷举搜索法。穷举搜索法是对可能是解的众多候选解按某种顺序进行逐一枚举和检验，并从中找出那些符合要求的候选解作为问题的解。

(4) 贪婪法。贪婪法是一种不追求最优解，只希望得到较为满意解的方法。贪婪法一般可以快速得到满意的解，因为它省去了为找最优解要穷尽所有可能而必须耗费的大量时间。贪婪法常以当前情况为基础作最优选择，而不考虑各种可能的整体情况，所以贪婪法不要回溯。

(5) 分治法。分治法是把一个复杂的问题分成两个或更多的相同或相似的子问题，再把子问题分成更小的子问题，直到最后子问题可以简单地直接求解，原问题的解即子问题的解的合并。

(6) 动态规划法。动态规划是一种在数学和计算机科学中使用的，用于求解包含重叠子问题的最优化问题的方法。其基本思想是，将原问题分解为相似的子问题，在求解的过程中通过子问题的解求出原问题的解。动态规划的思想是多种算法的基础，被广泛应用于计算机科学和工程领域。

(7) 迭代法。迭代法是数值分析中通过从一个初始估计出发寻找一系列近似解来解决问题（一般是解方程或者方程组）的过程，为实现这一过程所使用的方法统称为迭代法。

4. 良好的算法设计的要求

一个好的算法应达到如下目标：

(1) 正确性 (Correctness)。算法的计算结果必须要是正确的。

(2) 可读性 (Readability)。可读性好有助于用户对算法的理解；不易理解的程序易于隐藏较多错误，难以调试和修改。

(3) 健壮性 (Robustness)。当输入数据非法时，算法也能适当地做出反应或进行处理，而不会产生莫名其妙的输出结果。

(4) 效率与低存储量需求。效率指的是程序执行时，对于同一个问题如果有多个算法可以解决，执行时间短的算法效率高；存储量需求指算法执行过程中所需要的最大存储空间。

1.1.2 算法的复杂度

算法复杂度分为空间复杂度和时间复杂度。

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。同一个算法用不同的语言实现，或者用不同的编译程序进行编译，或者在不同的计算机上运行，效率均不同。

2. 算法的空间复杂度

算法的空间复杂度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中所需要的额外空间。

【例题 1】算法的空间复杂度是指_____。(2009 年 9 月)

A) 算法在执行过程中所需要的计算机存储空间

B) 算法所处理的数据量

C) 算法程序中的语句或指令条数

D) 算法在执行过程中所需要的临时工作单元数

【例题分析】

由以上定义得知，此题选A。

1.1.3 数据结构的定义

数据结构 (Data Structure) 是指相互之间存在一种或多种特定关系的数据元素的集合。

数据 (Data) 是对客观事物的符号表示，在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。

数据元素 (Data Element) 是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。

在一般情况下，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着某种关系（也就是连续），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。

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的关系的的数据元素集合的表示。通俗地说，数据结构是指带有结构的数据元素的集合。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。

数据的逻辑结果是对数据元素之间的逻辑关系的描述。它可以用一个数据元素的集合和定义在此集合中的若干关系来表示。用 D 表示数据元素的集合，用 R 来表示数据元素之间的前后件的关系。即一个数据结构可以表示为 $B=(D, R)$ ，其中 B 表示数据结构。这就是一个二元关系的表示方式。

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（也称为数据的物理结构）。

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（也就是前后件关系），在数据的存储结构中，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，常用的结构有顺序、链接、索引等存储结构，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。

1.1.4 数据结构的表示

数据结构的表示除了用二元关系表示外，还可以直观地用图形表示。

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。

在数据结构中，没有前件的结点称为根结点；没有后件的结点称为终端结点（也称为叶子结点）。

一个数据结构中的结点可能是动态变化的。根据需要或在处理过程中，可以在一个数据结构中增加一个新结点（称为插入运算），也可以删除数据结构中的某个结点（称为删除运算）。插入与删除是对数据结构的两种基本运算。除此之外，对数据结构的运算还有查找、分类、合并、分解、复制和修改等。

1.1.5 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，一般将数据结构分为两大类型：线性结构与非线性结构。

线性结构满足如下条件：

- (1) 有且只有一个根结点。
- (2) 每一个结点最多有一个前件，也最多有一个后件。

如果一个数据结构不是线性结构，称之为非线性结构。如果在一个数据结构中一个数据元素都没有，则称该数据结构为空。线性结构与非线性结构都可以是空的数据结构。对于空的数据结构，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。

1.2 线性表

本节主要考查线性表的基本概念，以及线性表的顺序存储方式。

1.2.1 线性表概述

线性表是一种常用的数据结构。

在实际应用中，线性表都是以栈、队列、字符串、数组等特殊线性表的形式来使用的。由于这些特殊线性表都具有各自的特性，因此，掌握这些特殊线性表的特性，对于数据运算的可靠性和提高操作效率都是至关重要的。

线性表是一个线性结构，它是一个含有 $n \geq 0$ 个结点的有限序列，对于其中的结点，有且仅有一个开始结点没有前驱但有一个后继结点，有且仅有一个终端结点没有后继但有一个前驱结点，其他结点都有且仅有一个前驱和一个后继结点。一般而言，一个线性表可以表示成一个线性序列： $k_1, k_2, \dots, k_n$ ，其中 k_1 是开始结点， k_n 是终端结点。

线性结构的基本特征如下：

- (1) 集合中必存在唯一的一个“第一元素”。
- (2) 集合中必存在唯一的一个“最后元素”。
- (3) 除最后一个元素之外，均有唯一的后继（后件）。
- (4) 除第一个元素之外，均有唯一的前驱（前件）。

由 n ($n \geq 0$) 个数据元素（结点） $a_1, a_2, \dots, a_n$ 组成的有限序列。

数据元素的个数 n 定义为表的长度。

当 $n=0$ 时称为空表。

常常将非空的线性表 ($n > 0$) 记做：

$$(a_1, a_2, \dots, a_n)$$

1.2.2 线性表的顺序存储

线性表的顺序存储指的是用一组地址连续的存储单元依次存储线性表的数据元素。线性表的顺序存储又叫做顺序表。

1. 线性表的顺序存储基本概念

线性表的顺序存储结构具备如下两个基本特征：

- (1) 线性表中的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。

假设线性表的每个元素需要占用 k 个存储单元，并以所占的存储位置 $ADR(a_{i+1})$ 和第 i 个数据元素的存储位置 $ADR(a_i)$ 之间满足下列关系：

$$ADR(a_{i+1}) = ADR(a_i) + k$$

线性表第 i 个元素 a_i 的存储位置为

$$\text{ADR}(a_i) = \text{ADR}(a_1) + (i-1) \times k$$

公式中 $\text{ADR}(a_1)$ 是线性表的第一个数据元素的存储位置, 通常称做线性表的起始位置或基址。

在 C 语言中, 通常定义一个一维数组来表示线性表的顺序存储空间, 如图 1-1 所示。

在用一维数组存放线性表时, 该一维数组的长度通常要定义得比线性表的实际长度大一些, 以便对线性表进行各种运算, 特别是插入运算。

a[0]	a[1]	a[2]	a[3]	a[4]
------	------	------	------	------

图 1-1

在线性表的顺序存储结构下, 可以对线性表做以下运算:

- (1) 在线性表的指定位置处加入一个新的元素 (也就是线性表的插入)。
- (2) 在线性表中删除指定的元素 (也就是线性表的删除)。
- (3) 在线性表中查找某个 (或某些) 特定的元素 (也就是线性表的查找)。
- (4) 对线性表中的元素进行排序 (也就是线性表的排序)。
- (5) 将一个线性表分解成多个线性表 (也就是线性表的分解)。
- (6) 将多个线性表合并成一个线性表 (也就是线性表的合并)。
- (7) 复制一个线性表 (也就是线性表的复制)。
- (8) 逆转一个线性表 (也就是线性表的逆转)。

2. 顺序表的基本操作

顺序表的基本操作包括插入运算和删除运算。

(1) 顺序表的插入运算

线性表的插入运算是指在表的第 i ($1 \leq i \leq n+1$) 个位置上, 插入一个新结点 x , 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

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

现在分析算法的复杂度。设它的值为 n 。该算法的时间主要花费在循环结点后移语句上, 该语句的执行次数 (也就是移动结点的次数) 是 $n-i+1$ 。由此可看出, 所需移动结点的次数不仅依赖于表的长度, 而且还与插入位置有关。

当 $i=n+1$ 时, 由于循环变量的终值大于初值, 结点后移语句将不进行; 这是最好的情况, 其时间复杂度为 $O(1)$ 。

当 $i=1$ 时, 结点后移语句, 将循环执行 n 次, 需移动表中的所有结点, 这是最坏的情况, 其时间复杂度为 $O(n)$ 。

综合以上的情况, 得出算法的平均时间复杂度为 $O(n)$ 。

(2) 顺序表的删除运算

线性表的删除运算是指将表的第 i ($1 \leq i \leq n$) 个结点删除, 使长度为 n 的线性表:

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

变成长度为 $n-1$ 的线性表

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

该算法的时间分析与插入算法相似, 结点的移动次数也是由表长 n 和位置 i 决定的。

若 $i=n$, 则由于循环变量的初值大于终值, 前移语句将不执行, 无须移动结点。

若 $i=1$, 则前移语句将循环执行 $n-1$ 次, 需移动表中除开始结点外的所有结点。这两种情况下算法的时间复杂度分别为 $O(1)$ 和 $O(n)$ 。

综合以上的情况得出在顺序表上做删除运算, 平均要移动表中约一半的结点, 平均时间复杂度也是 $O(n)$ 。

1.3 栈和队列

栈和队列都是特殊的线性表, 其定义符合线性表的定义, 其操作也类似于线性表的操作, 只不过增加了一些限定而已。

1.3.1 栈的定义与操作

栈 (Stack) 是一种特殊的线性表。栈是只能在表的一端进行插入和删除运算的线性表，通常称插入、删除的这一端为栈顶 (Top)，另一端为栈底 (Bottom)，如图 1-2 所示。当表中没有元素时称为空栈。栈顶元素总是后插入的元素，从而也是最先被删除的元素；栈底元素总是最先被插入的元素，从而也是最后才能被删除的元素。

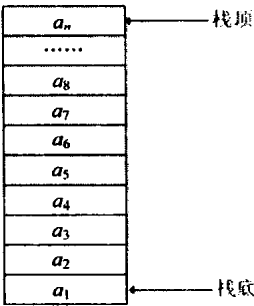


图 1-2

假设栈 $S=(a_1, a_2, a_3, \dots, a_n)$ ，则 a_1 称为栈底元素， a_n 为栈顶元素。栈中元素按 $a_1, a_2, a_3, \dots, a_n$ 的次序进栈，退栈的第一个元素应为栈顶元素。换句话说，栈的修改是按后进先出的原则进行的。因此，栈称为“先进后出”表 (FILO, First In Last Out)，或“后进先出”表 (LIFO, Last In First Out)。

栈的操作主要有入栈运算、退栈运算 (出栈) 和读栈顶元素。

(1) 入栈运算：入栈运算是指在栈顶位置插入一个新元素。首先将栈顶指针加一 (也就是 Top 加 1)，然后将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一个位置时，说明栈空间已满，不可能再进行入栈操作。这种情况称为栈“上溢”错误。

(2) 退栈运算：退栈是指取出栈顶元素并赋给一个指定的变量。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，然后将栈顶指针减一 (也就是 Top 减 1)。当栈顶指针为 0 时，说明栈空，不可进行退栈操作。这种情况称为栈的“下溢”错误。

(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。这个运算不删除栈顶元素，只是将它赋给一个变量，因此栈顶指针不会改变。当栈顶指针为 0 时，说明栈空，读不到栈顶元素。

1.3.2 队列的定义与操作

队列 (Queue) 是只允许在一端删除，在另一端插入的顺序表，允许删除的一端叫做队头 (Front)，允许插入的一端叫做队尾 (Rear)，如图 1-3 所示。

1. 队列的运算

当队列中没有元素时称为空队列。在空队列中依次加入元素 $a_1, a_2, \dots, a_n$ 之后， a_1 是队头元素， a_n 是队尾元素。显然退出队列的次序也只能是 $a_1, a_2, \dots, a_n$ ，也就是说队列的修改是依先进先出的原则进行的。因此队列亦称做“先进先出” (FIFO, First In First Out) 的线性表，或“后进后出” (LILO, Last In Last Out) 的线性表。

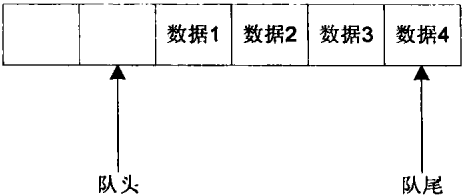


图 1-3

- (1) 入队操作。往队列队尾插入一个元素称为入队运算。
- (2) 出队操作。从队列队头删除一个元素称为出队运算。

2. 循环队列的运算

所谓循环队列，就是将队列存储空间的一个位置绕到第一个位置，形成逻辑上的环状空间。

在循环队列中，用队尾指针 Rear 指向队列中的队尾元素，用队头指针 Front 指向排头元素的前一个位置。因此，从排头指针 Front 指向的后一个位置直到队尾指针 Rear 指向的位置之间所有的元素均为队列中的元素。

在循环队列中进行出队、入队操作时，头尾指针仍要加 1，朝前移动。只不过当队尾指针指向向量上界 (Queuesize-1) 时，其加 1 操作的结果是指向向量的下界 0。

由于入队时尾指针向前追赶头指针，出队时头指针向前追赶尾指针，故队空和队满时头尾指针均相等。因此，我们无法通过 $Front=Rear$ 来判断队列是“空”还是“满”。在实际使用循环队列时，为了能区分队列满还是队列空，通常还需增加一个标志值的定义如下：当 $s=0$ 时表示队列空；当 $s=1$ 时表示队列非空。

(1) 入队运算。入队运算是指在循环队列的队尾加入一个新元素。首先将队尾指针进一 (也就是 $Rear=Rear+1$)，并当 $Rear=m+1$ 时置 $Rear=1$ ，然后将新元素插入到队尾指针指向的位置。当循环队列非空 ($s=1$) 且队尾指针等于队头指针时，说明循环队列已满，不能进行入队运算，这种情况称为“上溢”。

(2) 退队运算。退队运算是指在循环队列的队头位置退出一个元素并赋给指定的变量。首先将队头指针进一（也就是 $\text{Front}=\text{Front}+1$ ），并当 $\text{Front} = m+1$ 时，置 $\text{Front}=1$ ，然后将排头指针指向的元素赋给指定的变量。当循环队列为空（ $s=0$ ）时，不能进行退队运算，这种情况称为“下溢”。

【例题2】在下列数据结果中，能够按照“先进后出”原则存取数据的是_____。（2009年9月）

A) 循环队列

B) 栈

C) 队列

D) 二叉树

【例题分析】

按照先进后出的原则存取数据的是栈。

1.4 线性链表

本节主要考查线性链表的存储方式和基本操作。

1.4.1 线性表的链式存储

在定义的链表中，若只含有一个指针域来存放下一个元素地址，称这样的链表为单链表或线性链表。

在链式存储方式中，要求每个结点由两部分组成：一部分用于存放数据元素值，称为数据域；另一部分用于存放指针，称为指针域。其中，指针用于指向该结点的前一个或后一个结点（也就是前件或后件）。线性表的链式存储如图1-4所示。

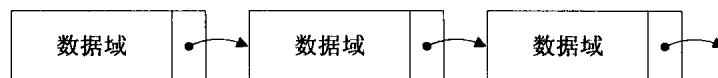


图 1-4

1. 线性链表的存储结构

用一组任意的存储单元来依次存放线性表的结点，这组存储单元既可以是连续的，也可以是不连续的，甚至是零散分布在内存中的任意位置上的。因此，链表中结点的逻辑次序和物理次序不一定相同。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，还必须存储指示其后件结点的地址（或位置）信息，这个信息称为指针（Pointer）或链（Link）。这两部分组成了链表中的结点结构。

链表通过每个结点的链域将线性表的 n 个结点按其逻辑次序链接在一起。由于上述链表的每一个结点只有一个链域，故将这种链表称为单链表（Single Linked）。

显然，单链表中每个结点的存储地址是存放在其前驱结点 Next 域中，而开始结点无前驱，故应设头指针 HEAD 指向开始结点。同时，由于终端结点无后件，故终端结点的指针域为空，即 NULL。

2. 线性链表的基本运算

线性链表的基本运算包括在线性链表中查找指定元素、在线性链表中插入元素、在线性链表中删除元素。

(1) 在线性链表中查找指定元素

在对线性链表进行插入或删除的运算中，总是首先需要找到插入或删除的位置，这就需要对线性链表进行扫描查找，在线性链表中寻找包含指定元素的前一个结点。

在链表中，查找是否有结点值等于给定值 x 的结点，若有的话，则返回首次找到的其值为 x 的结点的存储位置；否则返回 NULL。查找过程从开始结点出发，顺着链表逐个将结点的值和给定值 x 作比较。

(2) 在线性链表中插入元素

线性链表的插入是指在链式存储结构下的线性链表中插入一个新元素。

插入运算是将值为 x 的新结点插入到表的第 $i-1$ 个结点和第 i 个结点之间。因此，我们必须首先找到 a_{i-1} 的存储位置 p ，然后生成一个数据域为 x 的新结点 $*p$ ，并令结点 p 的指针域指向新结点，新结点的指针域指向结点 a_i 。

由线性链表的插入过程可以看出，由于插入的新结点取自于可利用栈，因此，只要可利用栈不空，在线性链表插入时总能取到存储插入元素的新结点，不会发生“上溢”的情况。而且，由于可利用栈是公用的，多个线性链表可以共享它，从而很方便地实现了存储空间动态分配。另外，线性链表在插入过程中不发生数据元素移动的现象，只要改变有关结点的指针即可，从而提高了插入的效率。

(3) 在线性链表中删除元素

线性链表的删除是指在链式存储结构下的线性链表中删除包含指定元素的结点。

删除运算是将表的第 i 个结点删去。因为在单链表中结点 a_i 的存储地址是在其直接前趋结点 a_{i-1} 的指针域 Next 中，所以必须首先找到 a_{i-1} 的存储位置 p 。然后令 $p \rightarrow \text{Next}$ 指向 a_i 的直接后继结点，即把 a_i 从链上删除，最后释放结点 a_i 的空间。

从线性链表的删除过程可以看出，从线性链表中删除一个元素后，不需要移动表中的数据元素，只要改变被删除元素所在结点的前一个结点的指针域即可。另外，可利用栈是用于收集计算机中所有的空闲结点，因此，当从线性链表中删除一个元素后，该元素的存储结点就变为空闲，应将空闲结点送回到可利用栈。

1.4.2 双向链表的结构及其基本运算

双向链表也叫双链表（如图 1-5 所示），是链表的一种，它的每个数据结点中都有两个指针，分别指向直接后继和直接前驱。所以，从双向链表中的任意一个结点开始，都可以很方便地访问它的前驱结点和后继结点。

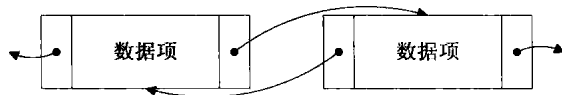


图 1-5

1. 双向链表的基本运算

(1) 插入：在 HEAD 为头指针的双向链表中，在值为 Y 的结点之后插入值为 X 的结点，插入结点的指针变化。

(2) 删除：在以 HEAD 为头指针的双向链表中删除值为 X 的结点，删除算法的指针变化。

2. 双向链表的结构及其基本运算

在前面所讨论的线性链表中，其插入与删除的运算虽然比较方便，但还存在一个问题，在运算过程中对于空表和对第一个结点的处理必须单独考虑，使空表与非空表的运算不统一。

因此，我们可以考虑建立这样的链表，具有单链表特征，但又不需要增加额外的存储空间，仅对表的链接方式稍做改变，使得对表的处理更加方便、灵活。从单链表可知，最后一个结点的指针域为 NULL，表示单链表已经结束。如果将单链表最后一个结点的指针域改为存放链表中头结点（或第一个结点）的地址，就使得整个链表构成一个环，又没有增加额外的存储空间，而满足这样条件的链表叫做循环链表。

循环链表具有以下两个特点：

(1) 在循环链表中增加了一个表头结点，其数据域为任意或者根据需要来设置，指针域指向线性表的第一个元素的结点。循环链表的头指针指向表头结点。

(2) 循环链表中最后一个结点的指针域不是空，而是指向表头结点。即在循环链表中，所有结点的指针构成了一个环状链。

在循环链表中，只要指出表中任何一个结点的位置，就可以从它出发访问到表中其他所有的结点，而线性单链表做不到这一点。

由于在循环链表中设置了一个表头结点，因此，在任何情况下，循环链表中至少有一个结点存在，从而使空表的运算统一。

1.5 树与二叉树

本节要求考生掌握树和二叉树的基本定义，重点考查二叉树的基本性质和二叉树的遍历。