

全国计算机技术与软件专业技术资格(水平)考试用书

新大纲

程序员考试 考点分析 与真题详解



施游王勇简亮
中国系统分析员顾问团
飞思教育产品研发中心

主编
组编
监制

飞思
考试中心



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

飞思考试中心

程序员考试考点分析与真题详解

施 游 王 勇 简 亮 主编

中国系统分析员顾问团 组编

飞思教育产品研发中心 监制

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书由中国系统分析员顾问团组织编写,作为全国计算机技术与软件专业技术资格(水平)考试中的程序员级别的考试用书。在参考和分析历年考试试题的基础上,着重对新版的考试大纲规定的内容有重点地细化和深化。本书内容涵盖了最新的程序员考试大纲(2004年修订版)的所有知识点,书中详尽分析和解答了1991—2004年的程序员考试试题。

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。准备参加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,熟悉考试方法、试题形式,试题的深度和广度,以及内容的分布,解答问题的方法和技巧等。

本书适合于广大程序员应试人员,也可作为程序员、软件设计师、数据库系统工程师、计算机专业教师教学和工作的参考书。

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

图书在版编目(CIP)数据

程序员考试考点分析与真题详解 / 施游, 王勇, 简亮主编. —北京: 电子工业出版社, 2005.2

(飞思考试中心)

ISBN 7-121-00762-2

I .程... II .①施...②王...③简... III.程序设计—工程技术人员—水平考试—自学参考资料
IV.TP311.1

中国版本图书馆CIP数据核字(2004)第141756号

责任编辑: 王树伟

印 刷: 北京东光印刷厂

出版发行: 电子工业出版社

北京海淀区万寿路173信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 36.5 字数: 934.5千字

印 次: 2005年2月第1次印刷

印 数: 8 000 册 定价: 53.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系电话: 010-68279077。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

全国计算机技术与软件专业技术资格（水平）考试是一个难度很大的考试，十多年来，考生平均通过率为 10% 左右。主要原因是考试范围十分广泛，牵涉到计算机专业的每门课程，还要加上数学，外语，系统工程，信息化和知识产权等知识，且注重考查新技术和新方法的应用。考试不但注重广度，而且还有一定的深度。特别是高级资格考试，不但要求考生具有扎实的理论知识，还要具有丰富的实践经验。

《程序员考试考点分析与真题详解》是为全国计算机技术与软件专业技术资格（水平）考试编写的考试用书，内容涵盖了最新的程序员考试大纲（2004 年修订版）的所有规定知识点（数学知识将另行出版书籍），书中详尽分析和解答了 1991-2004 年的程序员试题。

《程序员考试考点分析与真题详解》在参考和分析历年考试试题的基础上，着重对新版的考试大纲规定的内容有重点地细化和深化。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，熟悉考试方法、试题形式，试题的深度和广度，以及内容的分布，解答问题的方法和技巧等。

本书不仅对准备参加全国计算机技术与软件专业技术资格（水平）考试的读者有很大的作用，而且对从事软件设计工作的 IT 从业人员，计算机教学工作的老师，以及参加其它类似考试的读者也是有帮助的。

本书由中国系统分析员顾问团组编，由施游、王勇和简亮主编。万火、李雄、罗永红和王勇参加了审稿工作。

全书共分 16 章。第 1 章由王勇编写，第 2 章由施游编写，第 3 章由彭世强编写，第 4 章由殷建民编写，第 5 章由万火编写，第 6 章由罗永红编写，第 7 章由简亮编写，第 8 章由窦亚玲编写，第 9 章由张友生和陈建忠编写，第 10、11 章由张友生和徐锋编写，第 12 章由张友生、徐锋、田俊国和郑建兵编写，第 13 章由殷建民编写，第 14 章由郑睿和漆英编写，第 15、16 章由刘兴编写。

在本书出版之际，要特别感谢全国计算机技术与软件专业技术资格（水平）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。同时，本书在编写的过程中参考了许多相关的资料和书籍，在此恕不一一列举（详见参考文献列表），编者在此对这些参考文献的作者表示真诚的感谢。

由于编者水平有限，且本书涉及的知识点多，书中难免有不妥和错误之处，编者诚恳地期望各位专家和读者不吝指教和帮助，对此，我们将深为感激。

有关本书的意见反馈和咨询，可在中国系统分析员网站（<http://www.csai.cn>）“技术论坛”中的“CSAI 主编辅导教程”版块上与作者进行交流。

计算机技术与软件专业技术资格（水平）考试已走过了十几年历程，我们深感该考试对于推进国家信息化建设和软件产业化发展起着重要的作用。

编写委员会

目 录

第1章 数据结构与算法..... 1	2.1.1 机器语言与汇编语言.....60
1.1 算法设计概述.....1	2.1.2 汇编程序.....60
1.2 线性表.....3	2.1.3 装配程序.....62
1.2.1 栈.....5	2.1.4 宏指令.....63
1.2.2 队列.....5	2.2 编译系统基本原理.....64
1.2.3 稀疏矩阵.....6	2.2.1 编译概述.....64
1.2.4 字符串.....7	2.2.2 形式语言基本知识.....64
1.2.5 本节例题分析.....7	2.2.3 词法分析.....67
1.3 树和二叉树.....14	2.2.4 语法分析.....71
1.3.1 树.....14	2.2.5 语法制导翻译.....76
1.3.2 二叉树.....16	2.2.6 代码生成.....80
1.3.3 二叉排序树.....18	2.3 解释系统基本原理.....82
1.3.4 最优二叉树.....20	2.4 程序语言的数据类型.....84
1.3.5 本节例题分析.....22	2.4.1 基本数据类型.....84
1.4 图.....27	2.4.2 结构化数据类型.....87
1.4.1 图的基础知识.....27	2.4.3 抽象的数据类型.....88
1.4.2 最小生成树.....30	2.4.4 类型和错误检查.....89
1.4.3 最短路径.....30	2.5 程序语言的控制结构.....91
1.4.4 本节例题分析.....32	2.5.1 表达式.....91
1.5 排序与查找.....36	2.5.2 语句间的顺序控制.....93
1.5.1 插入排序.....36	2.5.3 过程控制.....95
1.5.2 选择排序.....37	2.6 本章例题分析.....97
1.5.3 交换排序.....37	第3章 操作系统基础知识..... 107
1.5.4 归并排序.....39	3.1 操作系统的功能、类型
1.5.5 基数排序.....40	和层次结构.....107
1.5.6 顺序查找.....41	3.2 处理机管理（进程管理）.....109
1.5.7 二分法查找.....41	3.3 存储管理.....113
1.5.8 分块查找.....43	3.4 设备管理.....116
1.5.9 本节例题分析.....43	3.5 文件管理.....118
1.6 递归法.....45	3.6 作业管理.....120
1.6.1 斐波纳契（Fibonacci）	3.7 网络操作系统.....122
数列.....46	3.8 嵌入式操作系统.....124
1.6.2 字典排序问题.....47	3.8.1 嵌入式操作系统概述.....124
1.6.3 本节例题分析.....49	3.8.2 常用的嵌入式
第2章 程序语言基础知识..... 59	操作系统.....125
2.1 汇编系统基本原理.....60	3.9 本章例题分析.....128

第4章 软件开发和运行维护

基础知识	141
4.1 软件工程和项目管理	
基础知识	141
4.1.1 软件工程基本概念	141
4.1.2 软件开发各阶段的目标和任务	142
4.1.3 软件过程基本知识	144
4.1.4 软件工程项目管理基本知识	147
4.1.5 面向对象开发方法基础知识	154
4.1.6 软件工具	156
4.1.7 软件开发环境	157
4.1.8 软件质量管理基础知识	158
4.1.9 软件开发模型	162
4.1.10 本节例题分析	164
4.2 系统分析与系统设计	
基础知识	170
4.2.1 结构化分析	170
4.2.2 面向对象的分析简介	173
4.2.3 系统设计的重要概念和基本原则	174
4.2.4 结构化设计	176
4.2.5 面向对象的设计	178
4.2.6 本节例题分析	179
4.3 程序设计基础知识	181
4.3.1 结构化程序设计及其工具	182
4.3.2 程序设计风格	183
4.3.3 面向对象设计基础知识、可视化程序设计基础知识	184
4.3.4 程序设计语言	184
4.3.5 本节例题分析	185
4.4 程序测试基础知识	188
4.4.1 黑盒测试、白盒测试和灰盒测试基础知识	188
4.4.2 测试工作流程	190
4.4.3 本节例题分析	192

4.5 软件开发文档基础知识	194
4.5.1 软件开发规范与文档标准	194
4.5.2 本节例题分析	196
4.6 软件运行和维护基础知识	196
4.6.1 软件运行基础知识	196
4.6.2 软件维护基础知识	198
4.6.3 本节例题分析	199

第5章 数据库系统

5.1 数据库管理系统的功能和特征	201
5.2 数据库模型	202
5.2.1 数据库系统的三级结构	202
5.2.2 数据库系统的三级模式	203
5.2.3 数据库系统两级独立性	204
5.3 数据模型	204
5.3.1 数据模型的分类	204
5.3.2 关系模型	205
5.3.3 E-R 模型图	206
5.4 数据操作	207
5.4.1 集合运算	207
5.4.2 关系运算	209
5.5 数据库语言	212
5.5.1 数据定义	212
5.5.2 数据查询	213
5.5.3 数据更新	217
5.5.4 视图	217
5.5.5 数据控制	219
5.6 数据库的控制功能	219
5.6.1 并发控制	219
5.6.2 数据恢复	221
5.6.3 安全性	222
5.6.4 完整性	225
5.7 本章例题分析	227
第6章 多媒体技术及其应用	241
6.1 多媒体技术基本概念	241
6.2 数据编码技术	242
6.2.1 数据编码方法	242

6.2.2 数据编码标准	245	7.8 本章例题分析	291
6.3 图形图像	249	第 8 章 存储器系统	305
6.4 音频	250	8.1 主存储器	305
6.5 视频	251	8.1.1 主要知识点	305
6.6 本章例题分析	252	8.1.2 本节例题分析	306
第 7 章 计算机硬件基础知识	257	8.2 辅助存储器	309
7.1 数制及其转换	257	8.2.1 磁带存储器	309
7.1.1 进制的表示法	257	8.2.2 磁盘存储器	309
7.1.2 进制的转换	257	8.2.3 RAID 存储器	310
7.2 数据的表示	258	8.2.4 光盘存储器	311
7.2.1 原码、反码、 补码、移码	258	8.2.5 存储网络	312
7.2.2 定点数和浮点数	259	8.2.6 本节例题分析	313
7.2.3 文字符号的编码	260	8.3 Cache 存储器	316
7.2.4 校验码概述	262	8.3.1 Cache 的实现	317
7.2.5 奇偶校验	262	8.3.2 Cache 的性能	320
7.2.6 海明码和恒比码	263	8.3.3 本节例题分析	321
7.3 算术运算和逻辑运算	264	第 9 章 安全性与系统性能指标	325
7.3.1 计算机中二进制数的 运算方法	264	9.1 数据安全与保密	325
7.3.2 逻辑代数的基本运算 和逻辑表达式的化简	266	9.1.1 数据加密算法	325
7.4 计算机系统的组成	269	9.1.2 身份认证技术	328
7.4.1 硬件和软件	269	9.1.3 信息网络安全协议	330
7.4.2 计算机语言	270	9.1.4 防火墙技术	332
7.4.3 冯·诺依曼结构	271	9.1.5 访问控制	333
7.5 计算机类型和特点	271	9.2 计算机病毒的防治	336
7.6 中央处理器 CPU	273	9.2.1 病毒的定义	336
7.6.1 CPU 的组成	273	9.2.2 病毒的特征	337
7.6.2 时序产生器 和控制方式	275	9.2.3 病毒的分类	339
7.6.3 计算机的分类	276	9.2.4 病毒的发展趋势	339
7.6.4 指令系统	277	9.2.5 病毒攻击的防范	341
7.6.5 处理器性能及评价	279	9.3 计算机木马的防治	342
7.7 输入/输出及通信设备	280	9.3.1 木马的定义	342
7.7.1 输入/输出设备一览	281	9.3.2 木马的特征	343
7.7.2 输入/输出控制器	281	9.3.3 木马的功能	343
7.7.3 外设的识别	282	9.3.4 木马的分类	344
7.7.4 外设的访问	283	9.3.5 木马的发展方向	345
7.7.5 常见输入/输出接口	287	9.3.6 防范木马的攻击	345
7.7.6 总线	289	9.3.7 木马的清除	347
		9.4 系统性能指标	347
		9.4.1 几个基本概念	347
		9.4.2 性能评价的常用指标 及方法	348

9.4.3	基准测试程序	350	11.4.1	专利法的保护对象	403
9.5	本章例题分析	351	11.4.2	确定专利权人	404
第 10 章	网络基础知识	355	11.4.3	专利权	405
10.1	网络的功能、分类与组成	355	11.5	反不正当竞争法	405
10.1.1	计算机网络的分类	355	11.5.1	什么是不正当竞争	405
10.1.2	按工作模式分类	357	11.5.2	商业秘密	406
10.1.3	计算机网络的组成	358	11.6	本章例题分析	407
10.2	网络协议与标准	364	第 12 章	标准化知识	411
10.2.1	OSI 网络层次模型	364	12.1	标准化概述	411
10.2.2	局域网协议	370	12.1.1	标准化的基本概念	411
10.2.3	广域网协议	373	12.1.2	标准化的发展	413
10.2.4	联网协议	375	12.2	标准的层次	415
10.3	网络结构与通信	376	12.3	编码标准	416
10.3.1	总线型拓扑结构	376	12.4	文件格式标准	418
10.3.2	星型拓扑结构	376	12.5	信息安全标准	418
10.3.3	环型拓扑结构	377	12.5.1	国际信息安全 等级标准	418
10.3.4	其他拓扑结构	378	12.5.2	国际信息技术 安全标准	419
10.3.5	拓扑结构的选择	378	12.5.3	中国的信息安全标准	421
10.4	三层结构	378	12.6	软件开发规范和文档标准	422
10.4.1	主机模式	378	12.7	标准化机构	423
10.4.2	客户/服务器模式	379	12.8	本章例题分析	424
10.4.3	三层结构与 B/S 模式	379	第 13 章	计算机专业英语	431
10.5	Internet 和 Intranet 初步	381	13.1	综述	431
10.5.1	Internet 网络协议	381	13.2	试卷分析	431
10.5.2	Internet 应用	385	13.3	例题详解	433
10.5.3	Intranet 初步	387	13.4	阅读素材	450
10.6	本章例题分析	387	13.5	计算机专业英语词汇 及缩略语精选	454
第 11 章	软件的知识产权保护	397	13.5.1	常见计算机词汇	455
11.1	著作权法及实施条例	397	13.5.2	常见计算机缩略语	461
11.1.1	著作权法客体	397	第 14 章	信息化基础知识	467
11.1.2	著作权法主体	398	14.1	信息与信息化	467
11.1.3	著作权	398	14.1.1	信息的定义及其特性	467
11.2	计算机软件保护条例	400	14.1.2	信息化	468
11.2.1	条例保护对象	400	14.1.3	信息化对组织的意义	469
11.2.2	著作权人确定	400	14.1.4	组织对信息化的需求	471
11.2.3	软件著作权	401	14.2	政府信息化与电子政务	472
11.3	商标法及实施条例	402	14.2.1	政府信息化的概念、 作用及意义	472
11.3.1	注册商标	402			
11.3.2	注册商标专用权保护	403			
11.3.3	注册商标使用的管理	403			
11.4	专利法及实施细则	403			

14.2.2	我国政府信息化的 历程和策略	473	15.1.1	信息系统的概念	511
14.2.3	电子政务的概念、内容 和技术形式	476	15.1.2	信息系统的功能	512
14.2.4	电子政务的应用领域 ..	479	15.1.3	信息系统的类型	515
14.2.5	电子政务建设的 过程模式 和技术模式	480	15.1.4	信息系统的发展	517
14.3	企业信息化与电子商务	483	15.2	信息系统建设	520
14.3.1	企业信息化的概念、 目的、规划、方法	483	15.2.1	信息系统建设的 复杂性	520
14.3.2	企业资源规划 (ERP) 的 结构和功能	486	15.2.2	信息系统的生命周期 ..	522
14.3.3	客户关系管理 (CRM) 在企业的应用	489	15.2.3	信息系统建设的原则 ..	524
14.3.4	企业门户	492	15.2.4	信息系统开发方法	525
14.3.5	企业应用集成	495	15.3	本章例题分析	529
14.3.6	供应链管理 (SCM) 的 思想	497	第 16 章	CSAI 模拟试题及试题分析 与解答	531
14.3.7	商业智能 (BI)	499	16.1	模拟试题——上午试题	531
14.3.8	电子商务的类型、 标准	502	16.2	模拟试题——下午试题	536
14.4	信息资源管理	504	16.3	模拟试题——上午试题分析 与解答	545
14.5	信息化的有关政策、法规 和标准	507	16.4	模拟试题——下午试题分析 与解答	550
14.6	本章例题分析	508	附录	程序员级考试大纲 (2004 年新版)	561
第 15 章	信息系统基础知识	511		考试范围	561
15.1	信息系统	511		考试科目 1: 计算机硬软件 基础知识	561
				考试科目 2: 程序设计	564
			参考文献		567

第 1 章 数据结构与算法

数据结构是指数据对象及其相互关系和构造方法，一个数据结构 S 可以用一个二元组表示为： $S=(D, R)$ 。其中， D 是数据结构中的数据的非空有限集合， R 是定义在 D 上的关系的非空有限集合。数据结构中，结点及结点间的相互关系称为数据的逻辑结构，数据在计算机中的存储形式称为数据的存储结构。

数据结构按逻辑结构不同分为线性结构和非线性结构两大类，其中非线性结构又可分为树形结构和图结构，而树形结构又可分为树结构和二叉树结构。

按照考试大纲的要求，在数据结构与算法方面，要求考生掌握以下知识点。

1. 常用数据结构

数组（静态数组、动态数组）、线性表、链表（单向链表、双向链表、循环链表）、队列、栈、树（二叉树、查找树）、图等的定义、存储和操作。

2. 常用算法

（1）排序算法、查找算法、数值计算算法、字符串处理算法、数据压缩算法、递归算法、图的相关算法。

（2）算法与数据结构的关系、算法效率、算法设计、算法描述（流程图、伪代码、决策表）、算法的复杂性。

1.1 算法设计概述

算法是在有限步骤内求解某一问题所使用的一组定义明确的规则。通俗地说，就是计算机解题的过程。在这个过程中，无论是形成解题思路还是编写程序，都是在实施某种算法。前者是推理实现的算法，后者是操作实现的算法。一个算法应该具有以下 5 个重要的特征：

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

（2）**确定性**：算法中每一条指令必须有确切的含义，读者理解时不会产生二义性。在任何条件下，算法只有惟一的一条执行路径，即对于相同的输入只能得出相同的输出。

（3）**输入**：一个算法有零个或多个的输入，以刻画运算对象的初始情况。所谓 0 个输入是指算法本身定出了初始条件。这些输入取自于某个特定的对象的集合。

（4）**输出**：一个算法有一个或多个输出，以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

（5）**可行性**：一个算法是可行的，即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。

算法设计要求正确性、可读性、健壮性、高效率与低存储量需求。

效率指的是算法执行时间。对于解决同一问题的多个算法，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。两者都与问题的规模有关。

算法的复杂性是算法效率的度量，是算法运行所需要的计算机资源的量，是评价算法优劣的重要依据。我们可以从一个算法的时间复杂度与空间复杂度来评价算法的优劣。当我们将一个算法转换成程序并在计算机上执行时，其运行所需要的时间取决于下列因素：

- (1) 硬件的速度。例如使用 486 还是使用 586。
- (2) 书写程序的语言。实现语言的级别越高，其执行效率就越低。
- (3) 编译程序所生成目标代码的质量。对于代码优化较好的编译程序其所生成的程序质量较高。
- (4) 问题的规模。例如，求 100 以内的素数与求 1000 以内的素数其执行时间必然是不同的。

显然，在各种因素都不能确定的情况下，很难比较出算法的执行时间。也就是说，使用执行算法的绝对时间来衡量算法的效率是不合适的。为此，可以将上述各种与计算机相关的软、硬件因素都确定下来，这样一个特定算法的运行工作量的大小就只依赖于问题的规模（通常用正整数 n 表示），或者说它是问题规模的函数。

1. 时间复杂度

一个程序的时间复杂度是指程序运行从开始到结束所需要的时间。

一个算法是由控制结构和原操作构成的，其执行时间取决于两者的综合效果。为了便于比较同一问题的不同的算法，通常的做法是：从算法中选取一种对于所研究的问题来说是基本运算的原操作，以该操作重复执行的次数作为算法的时间度量。一般情况下，算法中原操作重复执行的次数是规模 n 的某个函数 $T(n)$ 。

许多时候要精确地计算 $T(n)$ 是困难的，我们引入渐近时间复杂度在数量上估计一个算法的执行时间，也能够达到分析算法的目的。

定义（大 O 记号）：如果存在两个正常数 c 和 n_0 ，对于所有的 n ，当 $n \geq n_0$ 时有：

$$f(n) \leq cg(n)$$

则有：

$$f(n) = O(g(n))$$

也就是说，随着 n 的增大， $f(n)$ 渐近的不大于 $g(n)$ 。例如，一个程序的执行时间为 $T(n) = 2n^3 + n^2 + 5$ ，则 $T(n) = O(n^3)$ 。 $T(n)$ 和 n^3 的值随 n 的增大渐近地靠拢。

使用大 O 记号表示的算法的时间复杂度，称为算法的渐近时间复杂度。

通常用 $O(1)$ 表示常数计算时间。常见的渐近时间复杂度有：

$$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < O(2^n)$$

2. 空间复杂度

一个程序的空间复杂度是指程序运行从开始到结束所需的存储量。

程序运行所需的存储空间包括以下两部分：

固定部分：这部分空间与所处理数据的大小和个数无关。主要包括程序代码、常量、简单变量、定长成分的结构变量所占的空间。

可变部分：这部分空间大小与算法在某次执行中处理的特定数据的大小和规模有关。例如100个数据元素的排序算法与1000个数据元素的排序算法所需的存储空间显然是不同的。

算法由数据结构来体现，所以看一个程序首先要搞懂程序实现中所使用的数据结构，如解决装箱问题就使用链表这种数据结构。数据结构是算法的基础，数据结构支持算法，如果数据结构是递归的，算法也可以用递归来实现，如二叉树的遍历。经常采用的算法有迭代法、递推法、递归法、穷举法、贪婪法、分治法和回溯法等，根据考试大纲要求，程序员级别的考试，需要考生掌握递归法。

1.2 线 性 表

线性表是最简单和最常用的一种数据结构，线性表是由相同类型的结点组成的有限序列。一个由 n 个结点 $a_0, a_1, \dots, a_{n-1}$ 组成的线性表可记为 $(a_0, a_1, \dots, a_{n-1})$ 。线性表的结点个数为线性表的长度，长度为0的线性表称为空表。对于非空线性表， a_0 是线性表的第一个结点， a_{n-1} 是线性表的最后一个结点。线性表的结点构成一个序列，对序列中两相邻结点 a_i 和 a_{i+1} 称 a_i 是 a_{i+1} 的前驱结点， a_{i+1} 是 a_i 的后继结点。其中 a_0 没有前驱结点， a_{n-1} 没有后继结点。

线性表中结点之间的关系可由结点在线性表中位置所确定，通常用 (a_i, a_{i+1}) ($0 \leq i \leq n-2$) 表示两个结点之间的先后关系。例如，如果两个线性表有相同的数据结点，但它们的结点在线性表中出现的顺序不同，则它们是两个不同的线性表。

线性表的结点可由若干个成分组成，其中能惟一标识该结点的成分称为关键字，或简称键。为了讨论方便，往往只考虑结点的关键字，而忽略其他成分。

1. 线性表的基本运算

线性表包含的结点个数可以动态增加或减少，可以在任何位置插入或删除结点。线性表常用的运算可分成几类，每类有若干种运算。

(1) 查找运算

在线性表中查找具有给定键值的结点。

(2) 插入运算

在线性表的第 i ($0 \leq i \leq n-1$) 个结点的前面或后面插入一个新结点。

(3) 删除运算

删除线性表的第 i ($0 \leq i \leq n-1$) 个结点。

(4) 其他运算

- 统计线性表中结点的个数；
- 输出线性表各结点的值；
- 复制线性表；
- 线性表分拆；
- 线性表合并；
- 线性表排序；

- 按某种规则整理线性表。

2. 线性表的存储

线性表常用的存储方式有顺序存储和链接存储。

(1) 顺序存储

顺序存储是最简单的存储方式，通常用一个数组，从数组的第一个元素开始，将线性表的结点依次存储在数组中，即线性表的第 i 个结点存储在数组的第 i ($0 \leq i \leq n-1$) 个元素中，用数组元素的顺序存储来体现线性表中结点的先后次序关系。

顺序存储线性表的最大优点就是能随机存取线性表中的任何一个结点，缺点主要有两个：一是数组的大小通常是固定的，不利于任意增加或减少线性表的结点个数；二是插入和删除线性表的结点时，要移动数组中的其他元素，操作复杂。

(2) 链接存储

链接存储是用链表存储线性表（链表），最简单的是用单向链表，即从链表的第一个结点开始，将线性表的结点依次存储在链表的各结点中。链表的每个结点不但要存储线性表结点的信息，还要用一个域存储其后继结点的指针。单向链表通过链接指针来体现线性表中结点的先后次序关系。

链表存储线性表的优点是线性表的每个结点的实际存储位置是任意的，这给线性表的插入和删除操作带来方便，只要改变链表有关结点的后继指针就能完成插入或删除的操作，不需移动任何表元。链表存储方式的缺点主要有两个：一是每个结点增加了一个后继指针成分，要花费更多的存储空间；二是不方便随机访问线性表的任一结点。

3. 线性表上的查找

线性表上的查找运算是指在线性表中找某个键值的结点。根据线性表中的存储形式和线性表本身的性质差异，有多种查找算法，例如顺序查找、二分法查找、分块查找、散列查找等。其中二分法查找需要线性表是一个有序序列。

4. 在线性表中插入新结点

(1) 顺序存储

设线性表结点的类型为整型，插入之前有 n 个结点，把值为 x 的新结点插在线性表的第 i ($0 \leq i \leq n$) 个位置上。完成插入主要有以下步骤：

- 检查插入要求的有关参数的合理性；
- 把原来的第 $n-1$ 个结点至第 i 个结点依次往后移一个数组元素位置；
- 把新结点放在第 i 个位置上；
- 修改线性表的结点个数。

在具有 n 个结点的线性表上插入新结点，其时间主要花费在移动结点的循环上。若插入任一位置的概率相等，则在顺序存储性表中插入一个新结点，平均移动次数为 $n/2$ 。

(2) 链接存储

在链接存储线性表中插入一个键值为 x 的新结点，分以下 4 种情况：

- 在某指针 p 所指结点之后插入；
- 插在首结点之前，使待插入结点成为新的首结点；

- 接在线性表的末尾;
- 在有序链表中插入, 使新的线性表仍然有序。

5. 删除线性表的结点

(1) 顺序存储

在有 n 个结点的线性表中, 删除第 i ($0 \leq i \leq n-1$) 个结点。删除时应将第 $i+1$ 个表元至第 $n-1$ 个结点依次向前移一个数组元素位置, 共移动 $n-i-1$ 个结点。完成删除主要有以下几个步骤:

- 检查删除要求的有关参数的合理性;
- 把原来第 $i+1$ 个表元至第 $n-1$ 个结点依次向前移一个数组元素位置;
- 修改线性表表元个数。

在具有 n 个结点的线性表上删除结点, 其时间主要花费在移动表元的循环上。若删除任一表元的概率相等, 则在顺序存储线性表中删除一个结点, 平均移动次数为 $n/2$ 。

(2) 链接存储

对于链表中删除指定值结点的删除运算, 需考虑几种情况: 一是链表为空链表, 不执行删除操作; 二是要删除的结点恰为链表的首结点, 应将链表头指针改为指向原首结点的后继结点; 其他情况, 先要在链表中寻找要删除的结点, 从链表首结点开始顺序寻找。若找到, 执行删除操作, 若直至链表末尾没有指定值的结点, 则不实行删除操作。完成删除可有以下几个步骤组成:

- 如链表为空链表, 则不执行删除操作;
- 若链表的首结点的值为指定值, 更改链表的头指针为原首结点的后继结点;
- 在链表找指定值的结点;
- 将找到的结点删除。

1.2.1 栈

栈是一种特殊的线性表, 栈只允许在同一端进行插入和删除运算。允许插入和删除的一端称为栈顶, 另一端称为栈底。称栈的结点插入为进栈, 结点删除为出栈。因为最后进栈的结点必定最先出栈, 所以栈具有后进先出的特征。

1. 顺序存储

可以用顺序存储线性表来表示栈, 为了指明当前执行插入和删除运算的栈顶位置, 需要一个地址变量 top 指出栈顶结点在数组中的下标。

2. 链接存储栈

栈也可以用链表实现, 用链表实现的栈称为链接栈。链表的第一个结点为顶结点, 链表的首结点就是栈顶指针 top , top 为 NULL 的链表是空栈。

1.2.2 队列

队列也是一种特殊的线性表, 只允许在一端进行插入, 另一端进行删除运算。允许删

除运算的那一端称为队首, 允许插入运算的一端称为队尾。称队列的结点插入为进队, 结点删除为出队。因最先进入队列的结点将最先出队, 所以队列具有先进先出的特征。

1. 顺序存储

可以用顺序存储线性表来表示队列, 为了指明当前执行出队运算的队首位置, 需要一个指针变量 `head` (称为头指针), 为了指明当前执行进队运算的队尾位置, 也需要一个指针变量 `tail` (称为尾指针)。

若用有 N 个元素的数组表示队列, 随着一系列进队和出队运算, 队列的结点移向存放队列的数组的尾端, 会出现数组的前端空着, 而队列空间已用完的情况。一种可行的解决办法是当发生这样的情况时, 把队列中的结点移到数组的前端, 修改头指针和尾指针。另一种更好的解决办法是采用循环队列。

循环队列就是将实现队列的数组 $a[N]$ 的第一个元素 $a[0]$ 与最后一个元素 $a[N-1]$ 连接起来。队空的初态为 $head=tail=0$ 。在循环队列中, 当 `tail` 赶上 `head` 时, 队列满。反之, 当 `head` 赶上 `tail` 时, 队列变为空。这样队空和队满的条件都同为 $head=tail$, 这会给程序判别队空或队满带来不便。因此, 可采用当队列只剩下一个空闲结点的空间时, 就认为队列已满, 用这种简单办法来区别队空和队满。即对空的判别条件是 $head=tail$, 队满的判别条件是 $head=tail-1$ 。

2. 链接存储

队列也可以用链接存储线性表实现, 用链表实现的队列称为链接队列。链表的第一个结点是队列首结点, 链表的末尾结点是队列的队尾结点, 队尾结点的链接指针值为 `NULL`。队列的头指针 `head` 指向链表的首结点, 队列的尾指针 `tail` 指向链表的尾结点。当队列的头指针 `head` 值为 `NULL` 时, 队列为空。

1.2.3 稀疏矩阵

在计算机中存储一个矩阵时, 可使用二维数组。例如, $M \times N$ 阶矩阵可用一个数组 $a[M][N]$ 来存储 (可按照行优先或列优先的顺序)。如果一个矩阵的元素绝大部分为零, 则称为稀疏矩阵。若直接用一个二维数组表示稀疏矩阵, 则会因存储太多的零元素而浪费大量的内存空间。因此, 通常采用三元组数组或十字链表两种方法来存储稀疏矩阵。

1. 三元组数组

稀疏矩阵的每个非零元素用一个三元组来表示, 即非零元素的行号、列号和它的值。然后按某种顺序将全部非零元素的三元组存于一个数组中。

如果只对稀疏矩阵的某些单个元素作处理, 则宜用三元组表示。

2. 十字链表

在十字链表中, 矩阵的非零元素是一个结点, 同一行的结点和同一列的结点分别顺序循环链接, 每个结点既在它所在行的循环链表中, 又在它所在列的循环链表中。每个结点含 5 个域, 分别为结点对应的矩阵元素的行号、列号、值, 以及该结点所在行链表后继结点指针、所在列链表后继结点指针。

为了处理方便,通常对每个行链表和列链表分别设置一个表头结点,并使它们构成带表头结点的循环链表。为了引用某行某列的方便,全部行链表的表头结点和全部列链表的表头结点分别组成数组,这两个数组的首结点指针存于一个十字链表的头结点中,最后由一个指针指向该头结点。

如果对稀疏矩阵某行或某列整体作某种处理,可能会使原来为零的元素变为非零,而原来非零的元素可能变成零。对于这种场合,稀疏矩阵宜用十字链表来表示。

1.2.4 字符串

字符串是由某字符集上的字符所组成的任何有限字符序列。当一个字符串不包含任何字符时,称它为空字符串。一个字符串所包含的有效字符个数称为这个字符串的长度。一个字符串中任一连续的子序列称为该字符串的子串。

字符串通常存于足够大的字符数组中,每个字符串的最后一个有效字符之后有一个字符串结束标志,记为'\0'。通常由系统提供的库函数形成的字符串的末尾会自动添加'\0',但当由用户的应用程序来形成字符串时,必须由程序自行负责在最后一个有效字符之后添加'\0',以形成字符串。

对字符串的操作通常有:

- 统计字符串中有效字符的个数;
- 把一个字符串的内容复制到另一个字符串中;
- 把一个字符串的内容连接到另一个足够大的字符串的末尾;
- 在一个字符串中查找另一个字符串或字符;
- 按字典顺序比较两个字符串的大小。

1.2.5 本节例题分析

例题 1

设 W 为一个二维数组,其每个数据元素 W_{ij} 占用 6 个字节,行下标 i 从 0 到 8,列下标 j 从 2 到 5,则二维数组 W 的数据元素共占用 (A) 个字节。 W 中第 6 行的元素和第 4 列的元素共占用 (B) 个字节。若按行顺序存放二维数组 W ,其起始地址的字节号为 100,则二维数组 W 的最后一个数据元素的起始地址的字节号为 (C),数据元素 W_{34} 的起始地址号为 (D),而数据元素 W_{22} 的起始地址与当 W 按列顺序存放时数据元素 (E) 的起始地址相同。

供选择的答案

- | | | | |
|---------------|------------|------------|------------|
| A. ① 480 | ② 192 | ③ 216 | ④ 144 |
| B. ① 78 | ② 72 | ③ 66 | ④ 84 |
| C. ① 310 | ② 311 | ③ 315 | ④ 314 |
| D. ① 179 | ② 178 | ③ 184 | ④ 185 |
| E. ① W_{05} | ② W_{28} | ③ W_{52} | ④ W_{82} |

例题分析

已知 W 为一个二维数组,行下标 i 从 0 到 8,列下标 j 从 2 到 5,则 W 有 9 行 4 列共 36 个元素,如下所示:

W_{02}	W_{03}	W_{04}	W_{05}
W_{12}	W_{13}	W_{14}	W_{15}
W_{22}	W_{23}	W_{24}	W_{25}
W_{32}	W_{33}	W_{34}	W_{35}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
W_{82}	W_{83}	W_{84}	W_{85}

已知每个元素占 6 个字节, 则二维数组 W 的数据元素共占用 $4 \times 9 \times 6 = 216$ 个字节。

W 每一行有 4 个元素, 每一列有 9 个元素, 则 W 中第 6 行的元素和第 4 列的元素共有 12 个元素(其中 W_{64} 既属于第 6 行也属于第 4 列), 占用 $12 \times 6 = 72$ 个字节。

若按行顺序存放二维数组 W , 则首先把第 0 行的 4 个数据存放完毕, 然后再存放第 1 行的 4 个数据, 依此类推。题目告诉我们, 其起始地址的字节号为 100, 则二维数组 W 的最后一个数据元素的起始地址的字节号为 $100 + (36-1) \times 6 = 310$ 。数据元素 W_{34} 为第 3 行第 4 列, 其前面已经存储了 3 行再加 2 个元素(注意行从 0 起, 列从 2 起), 即 14 个元素, 因此 W_{34} 的起始地址号为 $100 + 14 \times 6 = 184$ 。

数据元素 W_{22} 为第 2 行第 2 列, 其前面已经存储了 2 行元素, 即 8 个元素。如果按列存储, 则先把第 2 列的 9 个数据存放完毕, 然后再存放第 3 列的 9 个数据, 依此类推。因此, 按行存储的 W_{22} 的起始地址与当 W 按列顺序存放时数据元素 W_{82} 的起始地址相同。

例题答案: A. ③ B. ② C. ① D. ③ E. ④

例题 2

设二维数组 F 的行下标为 1~5, 列下标为 0~8, F 的每个数据元素均占 4 个字节。在按行存储的情况下, 已知数据元素 $F[2,2]$ 的第一个字节的地址是 1044, 则 $F[3,4]$ 和 $F[4,3]$ 的第一个字节的地址分别为 (A) 和 (B), 而数组的第一个数据元素的第一个字节和数组最后一个元素的最后一个字节的地址分别为 (C) 和 (D)。

对一般的二维数组 G 而言, 当 (E) 时, 其按行存储的 $G[I,J]$ 的地址与按列存储的 $G[J,I]$ 的地址相同。

供选择的答案

- | | | | |
|--------------------------|----------------------------|--------|--------|
| A. ① 1088 | ② 1084 | ③ 1092 | ④ 1120 |
| B. ① 1092 | ② 1088 | ③ 1120 | ④ 1124 |
| C. ① 1004 | ② 1044 | ③ 1000 | ④ 984 |
| D. ① 1183 | ② 1179 | ③ 1164 | ④ 1187 |
| E. ① G 的列数与行数相同 | ② G 的列的上界与 G 的行的上界相同 | | |
| ③ G 的列的下界与 G 的行的下界相同 | ④ G 的列的上下界与 G 的行的上下界相同 | | |

例题分析

已知 F 为一个二维数组, 行下标为 1~5, 列下标为 0~8, 则 W 有 5 行 9 列共 45 个元素, 如下所示。

F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}
F_{20}	F_{21}	F_{22}	F_{23}	...				F_{28}
$\vdots$								$\vdots$
F_{50}	F_{51}	...						F_{58}

在按行存储的情况下, 数据元素 $F[2,2]$ 是存储第 $1 \times 9 + 3 = 12$ 个元素, $F[3,4]$ 是第 $2 \times 9 + 5 = 23$ 个元素, $F[4,3]$ 是第 $3 \times 9 + 4 = 31$ 个元素。已知每个元素占 4 个字节, $F[2,2]$ 第一个字节的地址是 1044, 则 $F[3,4]$ 的第一个字节的地址为 $1044 + (23-12) \times 4 = 1088$, $F[4,3]$ 的第一个字节的地址为 $1044 + (31-12) \times 4 = 1120$ 。

因为 $F[2,2]$ 第一个字节的地址是 1044, 其前面有 11 个元素共 $11 \times 4 = 44$ 个字节, 因此, 数组 F 的第一个数据元素的第一个字节的地址为 1000, 数组 F 最后一个元素的最后一个字节的地址为 $1000 + 45 \times 4 -$