

软件设计师考试

辅导教程

希赛教育软考学院 编著

考 点 突 破 · 案 例 分 析 · 实 战 练 习

帮助数万人通过软考的备战宝典全新升级!

迅速扣住考点 提升解题技巧 顺利通过考试



中国工信出版集团



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

全国计算机技术与软件专业技术资格

软件设计师考试

辅导教程



希赛教育软考学院 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书由希赛教育软考学院组织编写,作为软件设计师考试辅导的指定教材。全书内容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,掌握考试的重点和难点,熟悉内容的分布。

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

软件设计师考试辅导教程 / 希赛教育软考学院编著. —北京: 电子工业出版社, 2015.3
全国计算机技术与软件专业技术资格(水平)考试用书
ISBN 978-7-121-25614-1

I. ①软… II. ①希… III. ①软件设计—工程技术人员—资格考试—自学参考资料 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2015)第 041244 号

策划编辑: 孙学瑛

责任编辑: 徐津平

特约编辑: 赵树刚

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

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: 787×1092 1/16 印张: 26 字数: 749 千字

版 次: 2015 年 3 月第 1 版

印 次: 2015 年 3 月第 1 次印刷

定 价: 69.00 元

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

全国计算机技术与软件专业技术资格（水平）考试是由国家人力资源和社会保障部、工业和信息化部组织和领导的国家级考试，具有很高的权威性，但这同时也决定了其考试范围的广度和深度都比较大，使许多考生在复习和准备上都遇到了很多的难题。为帮助广大考生顺利通过考试，希赛教育软考学院组织编写了本书。

内容超值，针对性强

由于考试大纲规定的考试知识点体系庞大，对考生而言，要学习的内容很多。为此，希赛教育软考学院组织有关专家对考试大纲进行了深入的分析，在此基础上编写了本书，以作为计算机技术与软件专业技术资格（水平）考试中的软件设计师级别的考试辅导指定教材。

本书根据软件设计师的考试大纲编写而成，内容紧扣大纲，全面实用。本书在组织和写作上，倾注了作者们的许多精力和心血。相信本书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，全面进行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、教育产品开发和教育图书出版，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稳居国内首位，其远程教育模式得到了国家教育部门的认可和推广。

希赛教育软考学院（www.educity.cn/rk）是全国计算机技术与软件专业技术资格（水平）考试的顶级培训机构，拥有近 20 名资深软考辅导专家，负责考试大纲制定及软考辅导教材的编写工作。近年来共组织编写和出版了 100 多本软考教材，内容涵盖初级、中级和高级的各个专业，包括教程系列、辅导系列、考点分析系列、冲刺系列、串讲系列、试题精解系列、疑难解答系列、全程指导系列、案例分析系列、指定参考用书系列及一本通 11 个系列。希赛教育软考学院的专家录制了软考培训视频教程、串讲视频教程、试题讲解视频教程和专题讲解视频教程 4 个系列的软考视频。其软考教材、软考视频和软考辅导为考生助考并提高通过率做出了不可磨灭的贡献，在软考领域有口皆碑。特别是在高级资格领域，无论是考试教材，还是在线辅导和面授，希赛教育软考学院都独占鳌头。

本书由希赛教育软考学院的王勇主编，参加编写工作的还有张友生、谢顺、刘洋波、桂阳、胡光超、邓旭光、左水林、胡钊源、王军、王玉罡。

在线测试，心中有数

希赛网在线测试平台（www.educity.cn/tiku/）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”。这样系统就会自动显示考生原来做错的试题，供重新测试，以加强记忆。

考生可利用希赛网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的孙学瑛老师，她在本书的策划、选题的申报、写作大纲的确定，以及编辑和出版等方面付出了辛勤的劳动和智慧，给予我们很多的支持和帮助。

感谢参加希赛教育软考学院辅导和培训的学员，正是他们的想法汇成了本书的原动力，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，所以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。

希赛教育软考学院

2015年2月

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

传 真：(010) 88254397

E - m a i l: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目 录

第 1 章 数据结构基础	1	第 2 章 程序语言基础知识	35
1.1 线性表	1	2.1 汇编系统基本原理	36
1.1.1 栈	3	2.1.1 机器语言与汇编语言	36
1.1.2 队列	4	2.1.2 汇编程序	36
1.1.3 稀疏矩阵	4	2.2 编译系统基本原理	38
1.1.4 字符串	5	2.2.1 编译概述	38
1.2 树和二叉树	7	2.2.2 形式语言基本知识	39
1.2.1 树	7	2.2.3 词法分析	42
1.2.2 二叉树	9	2.2.4 语法分析	45
1.2.3 二叉排序树	11	2.2.5 语法翻译	46
1.2.4 平衡二叉树	13	2.2.6 代码生成	47
1.2.5 线索树	13	2.3 程序语言的控制结构	49
1.2.6 最优二叉树	13	2.3.1 表达式	49
1.3 图	15	2.3.2 语句间的顺序控制	51
1.3.1 图的基础知识	15	2.3.3 过程控制	53
1.3.2 最小生成树	18	2.4 程序语言的种类、特点及适用范围	54
1.3.3 最短路径	19	第 3 章 操作系统基础知识	56
1.3.4 拓扑排序	20	3.1 操作系统的功能、类型和层次结构	56
1.3.5 关键路径	21	3.2 处理机管理（进程管理）	57
1.4 排序	22	3.3 存储管理	62
1.4.1 插入排序	22	3.4 设备管理	64
1.4.2 选择排序	23	3.5 文件管理	66
1.4.3 交换排序	27	3.6 作业管理	69
1.4.4 归并排序	29	3.7 嵌入式操作系统	71
1.4.5 基数排序	29	第 4 章 软件工程基础知识	73
1.4.6 算法复杂性比较	31	4.1 软件生命周期与软件开发模型	73
1.5 查找	31	4.1.1 软件危机与软件工程	73
1.5.1 顺序查找	31	4.1.2 软件生命周期	74
1.5.2 二分法查找	32	4.1.3 软件开发模型	76
1.5.3 分块查找	33		
1.5.4 散列表	33		

4.2	主要软件开发方法	85	5.6.2	数据恢复	139
4.2.1	结构化分析和设计	85	5.6.3	安全性	140
4.2.2	面向数据结构的设计	88	5.6.4	完整性	143
4.2.3	面向对象的分析与设计	88	5.7	数据仓库与数据挖掘	145
4.3	软件测试与软件维护	93	5.7.1	数据仓库的概念	145
4.3.1	软件测试	93	5.7.2	数据仓库的结构	146
4.3.2	软件维护	102	5.7.3	数据挖掘技术概述	148
4.4	软件工具与软件开发环境	103	5.7.4	数据挖掘的功能	150
4.4.1	软件工具	103	5.7.5	数据挖掘常用技术	151
4.4.2	软件开发环境	104	5.7.6	数据挖掘的流程	152
4.5	软件质量保证	105	5.8	分布式数据库	153
4.5.1	软件质量	105	第 6 章	多媒体技术及其应用	156
4.5.2	软件质量特性	106	6.1	多媒体技术基本概念	156
4.5.3	软件质量保证	108	6.2	数据压缩标准	157
4.6	软件项目管理	111	6.3	图形图像	161
4.6.1	软件项目管理的内容	111	6.4	音频	162
4.6.2	软件项目估算	112	6.5	视频	164
4.6.3	软件项目组织与计划	114	第 7 章	计算机的体系结构和主要	165
4.6.4	风险管理	118		部件	165
第 5 章	数据库系统	119	7.1	机内代码及运算	165
5.1	数据库管理系统的功能和特征	119	7.1.1	数的进制	165
5.2	数据库模型	120	7.1.2	原码、反码、补码、移码	166
5.2.1	数据库系统的三级结构	120	7.1.3	定点数和浮点数	167
5.2.2	数据库系统的三级模式	121	7.1.4	校验码概述	169
5.2.3	数据库系统两级独立性	122	7.1.5	奇偶校验	169
5.3	数据模型	122	7.1.6	海明码	170
5.3.1	数据模型的分类	122	7.1.7	循环冗余校验码 (CRC)	170
5.3.2	关系模型	123	7.2	中央处理器 (CPU)	171
5.3.3	关系规范化理论	124	7.3	输入/输出控制方式	172
5.4	数据操作	126	7.4	指令流和数据流	177
5.4.1	集合运算	126	7.5	流水线技术	178
5.4.2	关系运算	128	7.5.1	流水线	178
5.5	数据库语言	130	7.5.2	影响流水线效率的因素	180
5.5.1	数据定义	131	7.6	精简指令计算机	181
5.5.2	数据查询	132	7.6.1	指令系统	181
5.5.3	数据更新	135	7.6.2	CISC 和 RISC	182
5.5.4	视图	135	第 8 章	存储器系统	184
5.5.5	数据控制	137	8.1	主存储器	184
5.6	数据库的控制功能	137			
5.6.1	并发控制	137			

8.2 辅助存储器	185	第 11 章 软件的知识产权保护	232
8.2.1 磁带存储器	185	11.1 著作权法及实施条例	232
8.2.2 磁盘存储器	185	11.1.1 著作权法客体	232
8.2.3 RAID 存储器	186	11.1.2 著作权法主体	233
8.2.4 光盘存储器	187	11.1.3 著作权	233
8.3 Cache 存储器	187	11.2 计算机软件保护条例	235
第 9 章 安全性、可靠性与系统性能		11.2.1 条例保护对象	235
评测	189	11.2.2 著作权人确定	235
9.1 数据安全与保密	189	11.2.3 软件著作权	235
9.1.1 数据加密算法	190	11.3 商标法及实施条例	237
9.1.2 身份认证技术	190	11.3.1 注册商标	237
9.1.3 信息网络安全协议	192	11.3.2 注册商标的专用权保护	237
9.1.4 防火墙技术	194	11.3.3 注册商标使用的管理	238
9.2 容错技术	195	11.4 专利法及实施细则	238
9.3 系统可靠性评价和系统性能评价方法	196	11.4.1 专利法的保护对象	238
9.3.1 系统可靠性评价的组合模型	196	11.4.2 确定专利权人	239
9.3.2 系统性能评价	198	11.4.3 专利权	239
第 10 章 网络基础知识	202	11.5 反不正当竞争法	240
10.1 网络的功能、分类与组成	202	11.5.1 不正当竞争	240
10.1.1 计算机网络的分类	202	11.5.2 商业秘密	241
10.1.2 按工作模式分类	203	第 12 章 计算机专业英语	242
10.1.3 计算机网络的组成	204	12.1 综述	242
10.2 网络协议与标准	210	12.2 计算机专业英语词汇及缩略语精选	242
10.2.1 OSI 网络层次参考模型	210	12.2.1 常见计算机词汇	243
10.2.2 局域网协议	216	12.2.2 常见计算机缩略语	249
10.2.3 广域网协议	216	第 13 章 信息化基础知识	255
10.2.4 互联网协议	219	13.1 信息与信息化	255
10.3 网络结构与通信	219	13.1.1 信息的定义及其特性	255
10.3.1 总线型拓扑结构	219	13.1.2 信息化	255
10.3.2 星型拓扑结构	219	13.1.3 组织对信息化的需求	256
10.3.3 环型拓扑结构	220	13.2 政府信息化与电子政务	257
10.3.4 其他拓扑结构	221	13.2.1 政府信息化的概念、作用	
10.3.5 拓扑结构的选择	221	及意义	258
10.4 Internet 和 Intranet 基础	221	13.2.2 我国政府信息化的历程和	
10.4.1 Internet 网络协议	221	策略	259
10.4.2 Internet 应用	227	13.2.3 电子政务的概念、内容和技	
10.4.3 Intranet 基础	229	术形式	260
10.5 网络管理基础	230	13.2.4 电子政务的应用领域	262

13.3 企业信息化与电子商务	263	16.3.3 面向对象设计	318
13.3.1 企业信息化的概念、目的、 规划、方法	263	16.4 用户界面设计	319
13.3.2 企业资源规划 (ERP) 的结构 和功能	266	16.5 设计评审	319
13.3.3 客户关系管理 (CRM) 在企业 的应用	269	第 17 章 数据流图设计	321
13.3.4 企业门户	272	17.1 数据流图	321
13.3.5 企业应用集成	274	17.1.1 数据流图基本图形符号	321
13.3.6 供应链管理 (SCM) 的思想	277	17.1.2 数据流图设计要略	322
13.3.7 商业智能 (BI)	279	17.1.3 数据字典	323
13.3.8 电子商务	282	17.1.4 分层数据流图	324
13.4 信息资源管理	283	17.1.5 分层数据流图的解答要点	324
第 14 章 信息系统基础知识	286	17.2 系统流程图	325
14.1 信息系统	286	17.2.1 系统流程图基本处理	325
14.1.1 信息系统的功能	287	17.2.2 系统流程图解题要点	326
14.1.2 信息系统的类型	290	第 18 章 UML 分析与设计	327
14.1.3 信息系统的发展	292	18.1 UML 概述	327
14.2 信息系统建设	295	18.1.1 UML 是什么	327
14.2.1 信息系统建设的复杂性	295	18.1.2 UML 结构	327
14.2.2 信息系统的生命周期	297	18.1.3 UML 的主要特点	329
14.2.3 信息系统建设的原则	299	18.1.4 UML 的应用领域	329
14.2.4 信息系统开发方法	300	18.2 用例图	329
第 15 章 标准化知识	304	18.2.1 用例基本概念	330
15.1 标准化概述	304	18.2.2 构建用例模型	331
15.2 标准的层次	305	18.2.3 用例的粒度	335
15.3 软件开发规范和文档标准	306	18.3 类图和对对象图	336
第 16 章 软件设计概述	308	18.3.1 类与类图的基本概念	336
16.1 软件设计基本原则	308	18.3.2 构建概念模型	339
16.1.1 信息隐蔽	308	18.3.3 类模型的发展	341
16.1.2 模块独立性	308	18.4 状态图	341
16.2 结构化设计方法	312	18.5 活动图	342
16.2.1 系统结构图中的模块	313	18.6 交互图	344
16.2.2 系统结构图中的主要成分	314	18.6.1 顺序图	344
16.2.3 常用的系统结构图	315	18.6.2 协作图 (通信图)	344
16.3 面向对象设计	317	18.7 构件图	345
16.3.1 面向对象的概念	317	18.8 包图	346
16.3.2 面向对象分析方法	318	18.9 部署图	347
		第 19 章 数据库设计	349
		19.1 数据的规范化	349
		19.1.1 函数依赖	349

19.1.2	码	350	19.5.2	设计用户子模式	374
19.1.3	1NF	350	19.5.3	数据模型优化	374
19.1.4	2NF	350	19.6	数据库物理设计	376
19.1.5	3NF	351	第 20 章	常用算法设计	378
19.1.6	BCNF	351	20.1	算法设计概述	378
19.1.7	逆规范化处理	351	20.2	递推法	380
19.2	数据库设计概述	351	20.3	递归法	381
19.3	需求分析	353	20.3.1	斐波那契 (Fibonacci) 数列	383
19.3.1	需求分析的任务	353	20.3.2	字典排序问题	383
19.3.2	确定设计目标	354	20.4	贪婪法	385
19.3.3	数据收集与分析	355	20.4.1	背包问题	386
19.3.4	需求说明书	355	20.4.2	装箱问题	390
19.4	概念结构设计	358	20.4.3	哈夫曼编码问题	393
19.4.1	概念结构	358	20.5	回溯法	397
19.4.2	概念结构设计的方法和步骤	359	20.5.1	组合问题	398
19.4.3	数据抽象和局部视图设计	359	20.5.2	子集和问题	400
19.4.4	局部 E-R 模型的集成	372	20.6	分治法	401
19.5	逻辑结构设计	372	20.7	动态规划法	403
19.5.1	E-R 图向关系模型的转换	373			

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。在数据结构中，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的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。

1.1 线性表

线性表是最简单、最常用的一种数据结构，它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的位置确定，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成分组成，其中能唯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中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了方便，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，如顺序查找、二分法查找、分块查找、散列查找等。其中二分法查找要求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。完成插入主要有如下步骤：

- 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。

- 修正线性表的结点个数。

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，则在顺序存储线性表中插入一个新结点，平均移动次数为 $(n-1)/2$ 。

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，分为如下 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 $i+1$ 个结点至第 $n-1$ 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。完成删除主要有如下几个步骤：

- 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。
- 修正线性表表元个数。

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

对于链表上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，则不执行删除操作。完成删除由如下几个步骤组成：

- 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指向首结点的后继结点。
- 在链表中寻找指定值的结点。
- 将找到的结点删除。

1.1.1 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。

1.1.2 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，就认为队列已满的简单办法，以区别队空和队满。即队空的判别条件是 $head=tail$ ，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。

1.1.3 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进行处理，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。

例如，矩阵 A 如图 1-1 所示。

$$\begin{bmatrix} 5 & 9 & 2 \\ 3 & 0 & 0 \\ 8 & 0 & 0 \end{bmatrix}$$

图 1-1 矩阵 A 示意图

则其十字链表如图 1-2 所示。

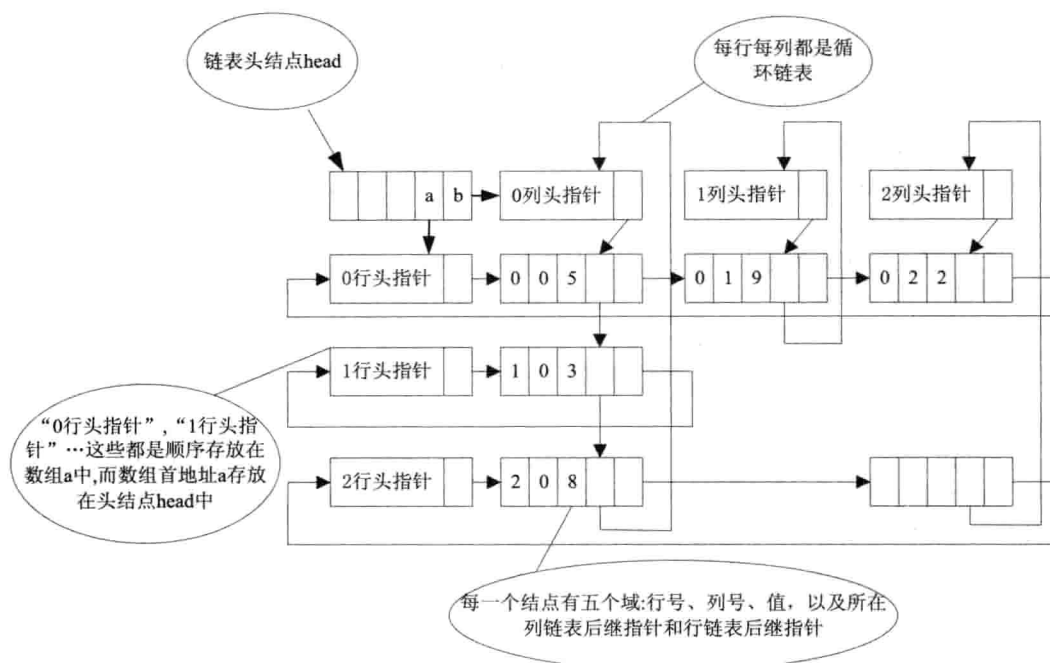


图 1-2 矩阵 A 十字链表存储示意图

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

1.1.4 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子序列称为该字符串的子串。

1. 字符串的存储

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。

2. 字符串的操作

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（字符串匹配）或字符。
- 按字典顺序比较两个字符串的大小。

3. KMP 算法

KMP 算法是一种改进的字符串匹配算法，由 D.E.Knuth 与 V.R.Pratt 和 J.H.Morris 同时发现，因此人们称它为克努特——莫里斯——普拉特操作（简称 KMP 算法）。KMP 算法相比传统的字符串匹配算法 BF 算法，有更高的效率，但其处理过程并不是很好理解。在软件设计师考试中，经常考查该算法中 next 数组的内容分析。

为了更好地理解 KMP 算法，首先要理解 BF 算法的操作过程。若要匹配 S 串中，有没有存在 T 串。则 BF 算法的核心思想是：首先 S[1]和 T[1]比较，若相等，则再比较 S[2]和 T[2]，一直到 T[M]为止；若 S[1]和 T[1]不等，则 S 的下标变更加 1，再依次进行比较，即 S[2]和 T[1]比较。如果存在 k，使 S[k+1...k+M]=T[1...M]，则匹配成功；否则失败。该算法效率表现最差的一面在于：当比较了 S[1]与 T[1]、S[2]与 T[2]、S[3]与 T[3]、S[4]与 T[4]，它们都相等，而 S[5]与 T[5]不相等时，BF 算法的做法是将 S 数组的下标回退到 2，将 S[2]与 T[1]对比，然后依次类推，这样做是非常低效的。所以该算法最坏情况下要进行 M*(N-M+1)次比较，时间复杂度为 O(M*N)。

在 KMP 算法中，主要改进的，也就是 BF 算法的劣势。KMP 算法中会构造一个 next 数组，这个数组用于记录，当某个字符匹配不成功时，接下来应该进行哪个元素的对比。下面以一个实例分析 next 数组是如何计算出来的。

例如，在字符串的 KMP 模式匹配法中，需要求解模式串 p 的 next 函数值，其定义如下。若模式串 p 为“aaabaaa”，则其 next 函数值为_____。

$$\text{next}[j] = \begin{cases} 0 & j = 1 \\ \max\{k | 1 < k < j, 'p_1p_2 \dots p_{k-1}' = 'p_{j-k+1}p_{j-k+2} \dots p_{j-1}'\} & \\ 1 & \text{其他情况} \end{cases}$$

- A. 0123123 B. 0123210 C. 0123432 D. 0123456

例题分析

KMP 模式匹配算法通俗来讲就是一种在一个字符串中定位另一个串的高效算法。其实在做这个题目时，也可以不需要知道 KMP 模式匹配算法，可以根据题目给出的定义式来求解。