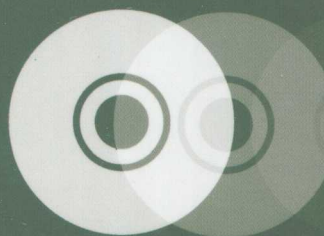


全国计算机等级考试

考点解析、例题精解
与实战练习



——二级C++语言 程序设计

NCRE研究组



高等教育出版社
Higher Education Press

全国计算机等级考试考点解析、例题精解与实战练习

——二级 C++ 语言程序设计

NCRE 研究组



高等教育出版社

内容提要

本书是按照教育部考试中心颁布的全国计算机等级考试最新考试大纲和指定教材编写的。全书分 17 章来讲解全国计算机等级考试二级 C++ 语言程序设计的知识, 章节安排与教育部考试中心指定的最新教材(2008 年版)同步, 主要从大纲要求、重要考点提示、经典题解和同步自测几个方面来对该部分内容做一个系统的阐释。涉及的内容主要有: 数据结构与算法, 程序设计基础, 软件工程基础, 数据库设计基础, C++ 语言概述, 数据类型、运算符和表达式, C++ 的基本语句和控制结构, 数组、指针与引用, 函数, 类和对象, 继承和派生, 运算符重载, 模板, C++ 流, 上机考试专题辅导, 笔试模拟试卷及答案分析, 机试模拟试卷及答案分析等。

本书具有考点浓缩、例题典型、讲解精当等特点, 非常适合参加全国计算机等级考试(二级 C++ 语言程序设计)的人员考前复习使用, 也适合其他相关人员使用。

本书配有光盘。盘中提供笔试和上机两大软件系统, 含有多套笔试与上机仿真模拟题。两大软件系统均增加了试题评析功能, 此外, 所有上机试题均配有视频演示及关键注解。

图书在版编目(CIP)数据

全国计算机等级考试考点解析、例题精解与实战练习.
二级 C++ 语言程序设计/NCRE 研究组. —北京: 高等教育出版社, 2008. 3

ISBN 978 - 7 - 04 - 023859 - 4

I. 全… II. N… III. ①电子计算机 - 水平考试 - 自学参考资料②C 语言 - 程序设计 - 水平考试 - 自学参考资料
IV. TP3

中国版本图书馆 CIP 数据核字 (2008) 第 028774 号

策划编辑	何新权	责任编辑	焦建虹	封面设计	张志奇	版式设计	陆瑞红
责任校对	朱惠芳	责任印制	宋克学				

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010 - 58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京新华印刷厂

开 本 787 × 1092 1/16
印 张 19.5
字 数 500 000

购书热线 010 - 58581118
免费咨询 800 - 810 - 0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2008 年 3 月第 1 版
印 次 2008 年 3 月第 1 次印刷
定 价 39.00 元(含光盘)

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 23859 - 00

前 言

全国计算机等级考试自 1994 年举办以来,得到了社会的广泛认可,在推广、普及计算机应用知识和技术,以及为用人单位录用和考核工作人员提供评价标准等方面发挥了重要的作用。全国计算机等级考试是目前国内参加人数最多、影响最大的计算机类考试。

为适应当前信息技术的飞速发展,教育部考试中心对全国计算机等级考试的考试科目及内容进行了重大调整,对考试大纲进行了全面修订。为了更好地服务于考生,引导考生尽快掌握先进的计算机技术,并顺利通过国家计算机等级考试,配合新考试大纲的推出,我们特别编写了本书。

本书分 17 章来讲解全国计算机等级考试二级 C++ 语言程序设计的知识,章节安排与教育部考试中心指定的最新教材(2008 年版)同步,主要从大纲要求、重要考点提示、经典题解和同步自测几个方面来对该部分内容做一个系统的阐释。涉及的内容主要有:数据结构与算法,程序设计基础,软件工程基础,数据库设计基础,C++ 语言概述,数据类型、运算符和表达式,C++ 的基本语句和控制结构,数组、指针与引用,函数,类和对象,继承和派生,运算符重载,模板,C++ 流,上机考试专题辅导,笔试模拟试卷及答案分析,机试模拟试卷及答案分析等。

与目前已出版的同类图书相比,本书具有如下特色:

1. 考点浓缩精解,重点突出。本书将指定的考试内容进行浓缩,用言简意赅的语言精讲考试要点、难点,从而使考生更易于理解全国计算机等级考试的要求和范围,能在较短时间内取得较大的收获。

2. 例题选取精心,分析到位。书中的例题一部分选自近年全国计算机等级考试的真题,一部分根据最新考试要求精心设计而成,具有典型性和针对性。所有例题均给出了详尽的分析,便于考生掌握完整的解题思路,以达到举一反三、触类旁通的功效。

3. 实战练习丰富,附有答案。本书针对考试过关,着重实战,每个章节均配有练习题,这些练习题是对逐段所学内容的巩固与提高,最后还提供了几套模拟试题,便于读者检测自己的总体水平。所有练习题、模拟题均配有答案,便于自测使用。

4. 上机全面辅导,书盘结合。书中介绍了上机考试的环境、要点、考试题型等内容,并对最新上机考试真题进行了分类详解。盘中提供笔试和上机两大软件系统,含有多套笔试与上机仿真模拟题,便于实战演练,让考生感受全真环境。两大软件均增加了试题评析功能,便于考生自我检查学习效果,在答题中加深对考核知识要点的理解与掌握。特别的,所有上机试题均配有视频演示及关键注解,如同名师亲临现场,手把手教考生如何解题过关。

本书非常适合参加全国计算机等级考试(二级 C++ 语言程序设计)的人员考前复习使用,也适合其他相关人员使用。

为方便考生复习,本书将重要考点或高频考点以“*”标记。

目 录

第1章 数据结构与算法

1.1 算法	1
考点1 算法的基本概念	1
*考点2 算法的复杂度	1
1.2 数据结构的基本概念	2
考点3 数据结构的定义	2
考点4 数据结构的图形表示	2
*考点5 线性结构与非线性结构	2
1.3 线性表及其顺序存储结构	3
考点6 线性表的基本概念	3
*考点7 线性表的顺序存储结构	3
考点8 顺序表的插入运算	3
考点9 顺序表的删除运算	3
1.4 栈和队列	4
*考点10 栈及其基本运算	4
*考点11 队列及其基本运算	4
1.5 线性链表	4
*考点12 线性链表的基本概念	4
考点13 线性链表的基本运算	4
考点14 循环链表及其基本运算	5
1.6 树与二叉树	5
考点15 树的基本概念	5
*考点16 二叉树及其基本性质	5
考点17 二叉树的存储结构	6
*考点18 二叉树的遍历	6
1.7 查找技术	6
*考点19 顺序查找	6
*考点20 二分法查找	6
1.8 排序技术	6
*考点21 交换类排序法	6
*考点22 插入类排序法	7
*考点23 选择类排序法	7
1.9 经典题解	7
1.10 同步自测	11

1.11 同步自测答案	11
-------------	----

第2章 程序设计基础

2.1 程序设计方法与风格	12
*考点1 程序设计方法与风格	12
2.2 结构化程序设计	13
*考点2 结构化程序设计的原则	13
*考点3 结构化程序的基本结构与特点	13
考点4 结构化程序设计原则和方法的应用	13
2.3 面向对象的程序设计	13
考点5 关于面向对象方法	13
*考点6 面向对象方法的基本概念	14
2.4 经典题解	14
2.5 同步自测	15
2.6 同步自测答案	16

第3章 软件工程基础

3.1 软件工程的基本概念	17
考点1 软件工程的定义	17
*考点2 软件生命周期的定义	17
考点3 软件工程的目标与原则	17
3.2 结构化分析方法	18
考点4 关于结构化分析方法	18
*考点5 关于结构化分析的常用工具	18
考点6 软件需求规格说明书	18
3.3 结构化设计方法	19
考点7 有关软件设计的基本内容	19
*考点8 有关结构化设计方法的基本内容	19
3.4 软件测试	20
*考点9 软件测试的方法与技术	20
*考点10 软件测试的实施	20
3.5 程序的调试	21

考点 11 程序调试的基本概念	21
* 考点 12 软件调试方法	21
3.6 经典题解	21
3.7 同步自测	24
3.8 同步自测答案	24

第 4 章 数据库设计基础

4.1 数据库系统的基本概念	25
* 考点 1 数据库	25
* 考点 2 数据库管理系统	25
* 考点 3 数据库系统	26
4.2 数据模型	26
考点 4 数据模型的基本概念	26
* 考点 5 E-R 模型 (实体-联系模型)	26
4.3 关系代数	27
考点 6 关系模型的基本操作	27
* 考点 7 关系模型的基本运算	27
* 考点 8 关系代数中的扩充运算	28
4.4 数据库设计与管理	28
考点 9 数据库设计方法与步骤	28
* 考点 10 数据库设计的需求分析	28
* 考点 11 数据库的概念设计	28
* 考点 12 数据库的逻辑设计	29
考点 13 数据库的物理设计	29
4.5 经典题解	29
4.6 同步自测	32
4.7 同步自测答案	33

第 5 章 C++ 语言概述

5.1 C++ 语言的发展和特点	34
考点 1 C++ 语言的发展	34
考点 2 C++ 语言的特点	34
5.2 面向对象程序设计	34
* 考点 3 面向对象程序设计	34
5.3 C++ 语言的基本符号和词汇	35
考点 4 C++ 语言的基本符号	35
考点 5 C++ 语言的五种词汇	35
5.4 C++ 程序的基本框架	37
考点 6 结构化程序设计框架	37
考点 7 面向对象程序设计框架	37
5.5 C++ 程序的开发过程	38

考点 8 用 Visual C++ 6.0 集成开发 环境编写和调度程序	38
5.6 经典题解	38
5.7 同步自测	40
5.8 同步自测答案	40

第 6 章 数据类型、运算符和表达式

6.1 C++ 语言的数据类型	41
* 考点 1 基本类型	41
* 考点 2 基本类型的派生类型	42
6.2 常量	42
* 考点 3 字面常量	42
* 考点 4 符号常量	44
6.3 变量的定义与使用方法	44
* 考点 5 变量的定义	44
考点 6 变量的使用方式	45
6.4 运算符和表达式	46
考点 7 运算符和表达式的概念	46
* 考点 8 赋值运算	46
* 考点 9 算术运算符和算术表达式	46
* 考点 10 关系运算符和关系表达式	47
* 考点 11 逻辑运算符和逻辑表达式	47
考点 12 位运算	47
考点 13 其他运算	48
* 考点 14 优先级和结合性	49
6.5 经典题解	50
6.6 同步自测	52
6.7 同步自测答案	52

第 7 章 C++ 的基本语句和控制结构

7.1 C++ 语句	53
考点 1 C++ 语句的分类	53
7.2 顺序结构	54
考点 2 声明语句	54
考点 3 表达式语句	54
考点 4 基本输入输出语句	55
考点 5 复合语句和空语句	55
7.3 选择结构	56
* 考点 6 基本 if 语句	56
* 考点 7 if...else 语句	56
* 考点 8 if 语句的嵌套	57

* 考点9 switch 语句	57	9.3 函数参数	84
7.4 循环结构	58	* 考点5 参数的传递方式	84
* 考点10 for 语句	58	* 考点6 默认参数	85
* 考点11 while 语句	59	9.4 函数重载	85
* 考点12 do...while 语句	59	* 考点7 函数重载的定义和使用	85
* 考点13 循环的嵌套	60	9.5 内联函数	85
* 考点14 混合嵌套	60	* 考点8 内联函数的概念	85
7.5 转向语句	60	9.6 递归函数	86
* 考点15 break 语句	60	* 考点9 递归函数的概念	86
考点16 continue 语句	60	9.7 变量的作用域与生存期	86
考点17 return 语句	61	考点10 存储类型	86
考点18 goto 语句	61	考点11 生存期	87
7.6 经典题解	61	9.8 经典题解	87
7.7 同步自测	65	9.9 同步自测	92
7.8 同步自测答案	68	9.10 同步自测答案	94
第8章 数组、指针与引用			
8.1 数组	69	10.1 类	95
* 考点1 一维数组	69	考点1 类的定义	95
* 考点2 二维数组	70	* 考点2 类成员的访问控制	96
考点3 多维数组	70	* 考点3 类的数据成员	96
* 考点4 字符数组	70	考点4 类的成员函数	97
8.2 指针	72	10.2 对象	97
* 考点5 指针和地址	72	* 考点5 对象的定义	97
* 考点6 指针和数组	73	* 考点6 对象成员及其访问	98
考点7 指针数组和函数指针	73	10.3 构造函数和析构函数	98
8.3 引用	74	* 考点7 构造函数和析构函数的定义	98
* 考点8 引用的基本概念和使用方法	74	考点8 默认构造函数和默认析构函数	99
8.4 动态存储分配	74	* 考点9 复制构造函数	99
* 考点9 动态内存空间	74	* 考点10 重载构造函数	100
8.5 经典题解	75	10.4 对象的生存期	100
8.6 同步自测	79	考点11 全局对象、静态对象与局部对象	100
8.7 同步自测答案	81	* 考点12 自由存储对象	100
第9章 函数			
9.1 函数的定义方法和调用方法	82	10.5 静态成员	101
* 考点1 函数定义	82	* 考点13 静态数据成员	101
* 考点2 函数调用	83	考点14 静态成员函数	101
9.2 函数原型和返回类型	83	10.6 常对象与常成员	102
考点3 函数原型	83	考点15 常对象	102
考点4 函数返回类型	84	考点16 常成员函数	102
		* 考点17 常数据成员	102

10.7 this 指针	103
* 考点 18 this 指针的使用	103
10.8 友元	103
* 考点 19 友元函数	103
考点 20 友元类	103
10.9 对象数组与成员对象	104
* 考点 21 对象数组	104
* 考点 22 成员对象	104
10.10 经典题解	105
10.11 同步自测	122
10.12 同步自测答案	131

第 11 章 继承和派生

11.1 继承和派生的概念	132
考点 1 派生类的基本概念	132
* 考点 2 派生类的定义与构成	133
11.2 派生类对基类成员的访问	133
* 考点 3 派生类对基类成员的访问 权限	133
11.3 派生类的构造函数和析构函数	134
* 考点 4 派生类的构造函数	134
* 考点 5 派生类的析构函数	135
11.4 多继承与虚基类	135
* 考点 6 多继承中的二义性问题	135
考点 7 虚基类的定义	135
考点 8 虚基类的构造函数	136
11.5 虚函数与多态性	136
考点 9 多态性的概念	136
* 考点 10 虚函数	136
* 考点 11 基类指针与派生类指针的 使用	137
* 考点 12 纯虚函数与抽象类	137
11.6 经典题解	137
11.7 同步自测	156
11.8 同步自测答案	166

第 12 章 运算符重载

12.1 运算符函数与运算符重载	167
考点 1 运算符重载的概念	167
* 考点 2 运算符重载的两种形式	168
12.2 典型运算符的重载	168
* 考点 3 典型运算符的重载	168

12.3 经典题解	170
12.4 同步自测	177
12.5 同步自测答案	181

第 13 章 模板

13.1 函数模板	182
* 考点 1 函数模板的概念和定义	182
* 考点 2 模板函数的实例化	183
* 考点 3 模板实参的省略	183
13.2 类模板	183
* 考点 4 类模板的概念和定义	183
* 考点 5 类模板的实例化	184
13.3 经典题解	184
13.4 同步自测	188
13.5 同步自测答案	191

第 14 章 C++ 流

14.1 C++ 流的概念	192
考点 1 C++ 流的体系结构	192
考点 2 预定义流对象	192
* 考点 3 提取运算符“>>”和插入运 算符“<<”	193
考点 4 有格式和无格式的输入输出	193
考点 5 操作符	193
14.2 输入输出的格式控制	194
* 考点 6 默认的输入输出格式	194
考点 7 格式控制标志	194
考点 8 格式控制函数	195
* 考点 9 输入输出宽度的控制	196
* 考点 10 浮点数输出方式的控制	196
* 考点 11 输出精度的控制	196
* 考点 12 对齐方式的控制	196
* 考点 13 小数点处理方式的控制	197
* 考点 14 填充字符的控制	197
* 考点 15 插入换行符	197
14.3 文件流	197
* 考点 16 文件流的建立、打开和关闭	197
考点 17 文件流状态的判别	198
考点 18 文件流的定位	199
考点 19 有格式的输入输出	200
14.4 经典题解	200
14.5 同步自测	206

14.6	同步自测答案	208
------	--------------	-----

第 15 章 上机考试专题辅导

15.1	数据类型的定义和使用	209
15.2	数组、指针的定义和使用	211
15.3	函数的定义和常用算法的函数实现	215
15.4	类和对象	223
15.5	继承和派生	229
15.6	运算符重载	235
15.7	模板的定义和使用	239
15.8	C++ 的流操作	243
15.9	同步自测	244
15.10	同步自测答案	253

第 16 章 笔试模拟试卷及答案分析

16.1	笔试模拟试卷	255
------	--------------	-----

16.1.1	笔试模拟试卷一	255
16.1.2	笔试模拟试卷二	261
16.1.3	笔试模拟试卷三	269
16.2	笔试模拟试卷答案分析	277
16.2.1	笔试模拟试卷一答案分析	277
16.2.2	笔试模拟试卷二答案分析	280
16.2.3	笔试模拟试卷三答案分析	283

第 17 章 机试模拟试卷及答案分析

17.1	机试模拟试卷	287
17.1.1	机试模拟试卷一	287
17.1.2	机试模拟试卷二	291
17.1.3	机试模拟试卷三	294
17.2	机试模拟试卷答案分析	298
17.2.1	机试模拟试卷一答案分析	298
17.2.2	机试模拟试卷二答案分析	299
17.2.3	机试模拟试卷三答案分析	300

第1章

数据结构与算法



大纲要求

- (1) 算法的基本概念，算法复杂度（时间复杂度与空间复杂度）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）。



重要考点提示

根据对历年真题的分析可知，本章考核内容约占15%，主要包括以下几个方面：

- (1) 算法复杂度。
- (2) 数据结构、栈、队列、线性链表的基本概念。
- (3) 二叉树的存储结构。
- (4) 线性表、树的结点的计算和遍历。
- (5) 冒泡排序的最坏次数计算。

1.1 算 法

考点1 算法的基本概念

- ① 作为一个算法，一般应具有4个基本特征：可行性、确定性、有穷性和拥有足够的信息。
- ② 一个算法通常由两种基本要素组成：一是对数据对象的运算和操作；二是算法的控制结构。算法的主要特征是着重于算法的动态执行。一个算法的功能不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。算法的控制结构不仅决定了算法中各操作的执行顺序，而且也直接反映了算法的设计是否符合结构化原则。

③ 算法设计的基本方法主要有列举法、归纳法、递推、递归、减半递推和回溯法。

*考点2 算法的复杂度

算法的复杂度主要包括时间复杂度和空间复杂度。

1. 算法的时间复杂度

所谓算法的时间复杂度，是指执行算法所需要的计算工作量。算法所执行的基本运算次数与问题的规模有关。算法的工作量用算法所执行的基本运算次数来度量。

对于一个固定的规模，算法所执行的基本运算次数还可能与特定的输入有关。在同一个问

题规模下,当算法执行所需的基本运算次数取决于某一特定输入时,可以用两种方法来分析算法的工作量:一种是平均性态分析,用 $A(n)$ 表示;另一种是讨论算法在最坏情况下的时间复杂度,用 $W(n)$ 表示,它比 $A(n)$ 更具有实用价值。

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用的空间、输入的初始数据所占用的存储空间以及算法执行过程中所需要的额外空间。在许多实际问题中,为了减少算法所占用的存储空间,通常采用压缩存储技术,以便尽量减少不必要的额外空间。

1.2 数据结构的基本概念

考点3 数据结构的定义

数据结构是指相互有关联的数据元素的集合。

1. 数据的逻辑结构

所谓数据的逻辑结构,是指反映数据元素之间逻辑关系的数据结构。数据的逻辑结构有两个要素:一是数据元素的集合,通常记为 D ;二是 D 上的关系,通常记为 R 。即一个数据结构可以表示成

$$B = (D, R)$$

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(也称数据的物理结构)。在进行数据处理时,被处理的各数据元素总是被存放在计算机的存储空间中,而且各数据元素在计算机存储空间中的位置关系与它们的逻辑关系可能不同。

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。而采用不同的存储结构,其数据处理的效率是不同的。

考点4 数据结构的图形表示

通常,一个数据结构中的元素结点可能是动态变化的。在对数据结构进行处理的过程中,不仅数据结构中的结点(即数据元素)的个数在动态地变化,而且各数据元素之间的关系也有可能动态地变化。

*考点5 线性结构与非线性结构

一般将数据结构分为线性结构与非线性结构。如果一个非空的数据结构满足两个条件,则称该数据结构为线性结构,这两个条件是:第一,有且只有一个根结点;第二,每一个结点最多有一个前件,也最多有一个后件。线性结构又称线性表。但特别需要说明的是,在一个线性结构中插入或删除任何一个结点后还应是线性结构。

如果一个数据结构不是线性结构,则称之为非线性结构。对非线性结构的存储与处理比对线性结构的存储和处理复杂得多。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是属于线性结构还是属于非线性结构,要根据具体情况来确定。

1.3 线性表及其顺序存储结构

考点6 线性表的基本概念

线性表由一组数据元素构成。矩阵是一个比较复杂的线性表。线性表是一个线性结构。数据元素在线性表中的位置只取决于它们自己的序号。

非空线性表有如下一些结构特征：

- ① 有且只有一个根结点，它无前件。
- ② 有且只有一个终端结点，它无后件。
- ③ 除根结点与终端结点外，其他所有结点有且只有一个前件，也有且只有一个后件。

考点7 线性表的顺序存储结构

线性表的顺序存储结构具有以下两个基本特点：

- 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。

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。在一般情况下，如果线性表的长度在处理过程中是动态变化的，则在开辟线性表的存储空间时要考虑到线性表在动态变化过程中可能达到的最大长度。

考点8 顺序表的插入运算

在线性表采用顺序存储结构时，如果插入运算在线性表的末尾进行，即在第 n 个元素之后（可以认为是在第 $n+1$ 个元素之前）插入新元素，则只要在表的末尾增加一个元素即可，不需要移动表中的元素；如果要在线性表的第 1 个元素之前插入一个新元素，则需要移动表中的所有元素。在一般情况下，如果插入运算在第 i ($1 \leq i \leq n$) 个元素之前进行，则原来第 i 个元素之后（包括第 i 个元素）的所有元素都必须移动。在平均情况下，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时更为突出。

考点9 顺序表的删除运算

在一般情况下，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 个元素之前共 $n-i$ 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。

在线性表采用顺序存储结构时，如果删除运算在线性表的末尾进行，即删除第 n 个元素，则不需要移动表中的元素；如果要删除线性表中的第 1 个元素，则需要移动表中的所有元素。在一般情况下，如果要删除第 i ($1 \leq i \leq n$) 个元素，则原来第 i 个元素之后的所有元素都必须依次向前移动一个位置。在平均情况下，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。

1.4 栈和队列

*考点 10 栈及其基本运算

栈是限定在一端进行插入与删除的线性表。栈顶元素是最后被插入的元素，从而也是最先能被删除的元素；栈底元素是最先被插入的元素，从而也是最后才能被删除的元素。即栈是按照“先进后出”或“后进先出”的原则组织数据的。

栈的基本运算有三种：入栈、退栈与读栈顶元素。

*考点 11 队列及其基本运算

队列是指允许在一端进行插入、而在另一端进行删除的线性表。队列又称为“先进先出”或“后进后出”的线性表，它体现了“先来先服务”的原则。在队列中，队尾指针与排头指针共同反映了队列中元素动态变化的情况。

在循环队列结构中，当存储空间最后一个位置已被使用而要再进行入队运算时，只要存储空间的第一个位置空闲，便可将元素加入到第一个位置，即将存储空间的第一个位置作为队尾。循环队列的初始状态为空。循环队列主要有两种基本运算：入队运算与退队运算。每进行一次入队运算，队尾指针就进一；每进行一次退队运算，排头指针就进一。

1.5 线性链表

*考点 12 线性链表的基本概念

在链式存储结构中，存储数据结构的存储空间可以不连续，各数据结点的存储顺序与数据元素之间的逻辑关系可以不一致，而数据元素之间的逻辑关系是由指针域来确定的。链式存储方式既可用于表示线性结构，也可用于表示非线性结构。

一般来说，在线性表的链式存储结构中，各数据结点的存储序号是不连续的，并且各结点在存储空间中的位置关系与逻辑关系也不一致。对于线性链表，可以从头指针开始，沿各结点的指针扫描到链表中的所有结点。

考点 13 线性链表的基本运算

1. 在线性链表中查找指定元素

在非空线性链表中寻找包含指定元素值 x 的前一个结点 p 的基本方法为：从头指针指向的结点开始往后沿指针进行扫描，直到后面已没有结点或下一个结点的数据域为 x 为止。因此，由这种方法找到的结点 p 有两种可能：当线性链表中存在包含元素 x 的结点时，则找到的 p 为第一次遇到的包含元素 x 的前一个结点序号；当线性链表中不存在包含元素 x 的结点时，则找到的 p 为线性链表中的最后一个结点数。

2. 线性链表的插入

为了要在线性链表中插入一个新元素，首先要给该元素分配一个新结点，以便用于存储该元素的值。新结点可以从可利用栈中取得。然后将存放新元素值的结点链接到线性链表中的指

定位置。多个线性链表可以共享栈，从而很方便地实现了存储空间的动态分配。线性链表在插入过程中不发生数据元素移动的现象，只需改变有关结点的指针即可，从而提高插入的效率。

3. 线性链表的删除

为了在线性链表中删除包含指定元素的结点，首先要在线性链表中找到这个结点，然后将要删除结点放回到可利用栈。在线性链表中删除一个元素后，不需要移动表中的数据元素，只需改变被删除元素所在结点的前一个结点的指针域即可。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，应将该空闲结点送回到可利用栈。

考点 14 循环链表及其基本运算

循环链表与前面所讨论的线性链表相比，其结构具有以下两个特点：

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。

1.6 树与二叉树

考点 15 树的基本概念

树是一种简单的非线性结构。在树这种数据结构中，所有数据元素之间的关系具有明显的层次特性。

树在计算机中通常用多重链表表示。多重链表中各结点的链域个数不同，这将导致对树进行处理的算法很复杂。如果用定长的结点来表示树中的每个结点，即取树的度作为每个结点的链域个数，这就可以使对树的各种处理算法大大简化。但在这种情况下，会造成存储空间的浪费，因为有可能在很多结点中存在空链域。

*考点 16 二叉树及其基本性质

二叉树具有以下两个特点：

① 非空二叉树只有一个根结点。

② 每个结点最多有两棵子树，且分别称为该结点的左子树与右子树。

二叉树有以下几个性质：

① 在二叉树的第 k 层上，最多有 2^{k-1} ($k \geq 1$) 个结点。

② 深度为 m 的二叉树最多有 $2^m - 1$ 个结点。

③ 在任意一棵二叉树中，度为 0 的结点总是比度为 2 的结点多一个。

④ 具有 n 个结点的二叉树，其深度至少为 $\lfloor \log_2 n \rfloor + 1$ ，其中 $\lfloor \log_2 n \rfloor$ 表示取 $\log_2 n$ 的整数部分。

在满二叉树中，每一层上的结点数都达到最大值，即在满二叉树的第 k 层上有 2^{k-1} 个结点，且深度为 m 的满二叉树有 $2^m - 1$ 个结点。

满二叉树是完全二叉树，而完全二叉树一般不是满二叉树。

考点 17 二叉树的存储结构

在计算机中, 二叉树通常采用链式存储结构。用于存储二叉树结点的存储地址的指针域有两个: 一个用于指向该结点的左子结点的存储地址, 称为左指针域; 另一个用于指向该结点的右子结点的存储地址, 称为右指针域。

***考点 18 二叉树的遍历**

二叉树的遍历是指不重复地访问二叉树中的所有结点。

在遍历二叉树的过程中, 一般先遍历左子树, 然后再遍历右子树。在先左后右的原则下, 根据访问根结点的次序, 二叉树的遍历可以分为三种: 前序遍历、中序遍历和后序遍历。下面介绍这三种遍历的方法。

1. 前序遍历

若二叉树为空, 则结束返回。否则: a. 访问根结点; b. 前序遍历左子树; c. 前序遍历右子树。

2. 中序遍历

若二叉树为空, 则结束返回。否则: a. 中序遍历左子树; b. 访问根结点; c. 中序遍历右子树。

3. 后序遍历

若二叉树为空, 则结束返回。否则: a. 后序遍历左子树; b. 后序遍历右子树; c. 访问根结点。

1.7 查找技术

***考点 19 顺序查找**

在进行顺序查找的过程中, 如果线性表中的第一个元素就是被查找元素, 则只需做一次比较就查找成功, 查找效率最高; 但如果被查找的元素是线性表中的最后一个元素, 或者被查找元素根本不在线性表中, 则为了查找这个元素需要与线性表中的所有元素进行比较, 这是顺序查找的最坏情况。在平均情况下, 利用顺序查找法在线性表中查找一个元素, 大约要与线性表中一半的元素进行比较。

***考点 20 二分法查找**

二分法查找只适用于顺序存储的有序表。在此所说的有序表是指线性表中的元素按值非递减排列 (即从小到大, 但允许相邻元素值相等)。二分法查找的效率要比顺序查找的效率高出得多。可以证明, 对于长度为 n 的有序线性表, 在最坏情况下, 二分查找只需要比较 $\log_2 n$ 次, 而顺序查找需要比较 n 次。

1.8 排序技术

***考点 21 交换类排序法**

假设线性表的长度为 n , 则在最坏情况下, 冒泡排序需要经过 $n/2$ 遍的从前往后的扫描和

$n/2$ 遍的从后往前的扫描, 需要的比较次数为 $n(n-1)/2$ 。但这个工作量不是必需的, 一般情况下要小于这个工作量。

快速排序法的关键是对线性表进行分割, 以及对各分割出的子表再进行分割。随着对各子表不断分割, 划分出的子表会越来越多, 但一次又只能对一个子表进行再分割处理, 需要将暂时不分割的子表记忆起来, 这就要用一个栈来实现。

*考点 22 插入类排序法

在简单插入排序法中, 每一次比较后最多移掉一个逆序, 因此, 这种排序方法的效率与冒泡排序法的效率相同。在最坏情况下, 简单插入排序需要 $n(n-1)/2$ 次比较。

希尔排序的效率与所选取的增量序列有关。

*考点 23 选择类排序法

简单选择排序法在最坏情况下需要比较 $n(n-1)/2$ 次。

堆排序的方法对于规模较小的线性表并不适合, 但对于较大规模的线性表来说是很有效的。在最坏情况下, 堆排序需要比较的次数为 $O(n\log_2 n)$ 。

1.9 经典题解

一、选择题

1. 下列叙述中, 正确的是_____。(2007 年 9 月)

- A. 程序执行的效率与数据的存储结构密切相关
- B. 程序执行的效率只取决于程序的控制结构
- C. 程序执行的效率只取决于所处理的数据量
- D. 以上三种说法都不对

解析: 采用不同的存储结构, 其数据处理的效率是不同的, 程序执行的效率也就不同, 因此选项 A 是正确的。程序执行的效率不仅与程序的控制结构有关, 而且与所处理的数据量有关, 因此选项 B、C、D 都不正确。

答案: A

2. 下列叙述中, 正确的是_____。(2007 年 9 月)

- A. 数据的逻辑结构与存储结构必定是一一对应的
- B. 由于计算机存储空间是向量式的存储结构, 因此, 数据的存储结构一定是线性结构
- C. 程序设计语言中的数组一般是顺序存储结构, 因此, 利用数组只能处理线性结构
- D. 以上三种说法都不对

解析: 数据元素在计算机存储空间中的位置关系与它们的逻辑关系不一定是相同的, 即数据的逻辑结构与存储结构不一定一一对应。

答案: D

3. 冒泡排序在最坏情况下的比较次数是_____。(2007 年 9 月)

- A. $n(n+1)/2$
- B. $n\log_2 n$
- C. $n(n-1)/2$
- D. $n/2$

解析: 假设线性表的长度为 n , 则在最坏情况下, 冒泡排序需要经过 $n/2$ 遍从前往后的扫描和 $n/2$ 遍从后往前的扫描, 需要的比较次数为 $n(n-1)/2$ 。

答案: C

4. 在一棵二叉树中共有 70 个叶子结点与 80 个度为 1 的结点, 则该二叉树中的总结点数为_____。(2007 年 9 月)

- A. 219
- B. 221
- C. 229
- D. 231

解析: 在任意一棵二叉树中, 度为 0 的结点总是比度为 2 的结点多一个。度为 2 的结点数为 69, 故总结

点数为 $69 + 70 + 80 = 219$ 。

答案: A

5. 下列叙述中, 正确的是_____。(2007年4月)

- A. 算法的效率只与问题的规模有关, 而与数据的存储结构无关
- B. 算法的时间复杂度是指执行算法所需要的计算工作量
- C. 数据的逻辑结构与存储结构是一一对应的
- D. 算法的时间复杂度与空间复杂度一定相关

解析: 算法的效率不仅与问题的规模有关, 而且还与数据的存储结构有关。算法的时间复杂度是指执行算法所需要的计算工作量。各数据元素在计算机存储空间中的位置关系与它们的逻辑关系可能不同。算法的时间复杂度与空间复杂度不一定相关。

答案: B

6. 下列对队列的叙述, 正确的是_____。(2007年4月)

- A. 队列属于非线性表
- B. 队列按“先进后出”的原则组织数据
- C. 队列在队尾删除数据
- D. 队列按“先进先出”的原则组织数据

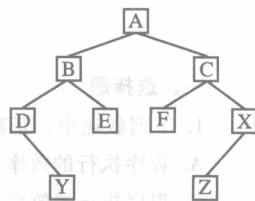
解析: 队列是指允许在一端进行插入、而在另一端进行删除的线性表。在队列这种数据结构中, 最先插入的元素将最先被删除, 反之, 最后插入的元素将最后被删除。因此, 队列又称为“先进先出”或“后进后出”的线性表, 它体现了“先来先服务”的原则。

答案: D

7. 对右图所示的二叉树进行前序遍历的结果为_____。(2007年4月)

- A. DYBEAFCZX
- B. YDEBFZXCA
- C. ABDYECFXZ
- D. ABCDEFXYZ

解析: 前序遍历首先访问根结点, 然后遍历左子树, 最后遍历右子树; 并且, 在遍历左、右子树时, 仍然先访问根结点, 然后遍历左子树, 最后遍历右子树。前序遍历的结果为 ABDYECFXZ。



答案: C

8. 某二叉树中有 n 个度为 2 的结点, 则该二叉树中的叶子结点数为_____。(2007年4月)

- A. $n+1$
- B. $n-1$
- C. $2n$
- D. $n/2$

解析: 在任意一棵二叉树中, 度为 0 的结点总是比度为 2 的结点多一个。故度为 0 的结点数为 $n+1$ 。

答案: A

9. 下列叙述中, 正确的是_____。(2006年9月)

- A. 一个算法的空间复杂度大, 则其时间复杂度也必定大
- B. 一个算法的空间复杂度大, 则其时间复杂度必定小
- C. 一个算法的时间复杂度大, 则其空间复杂度必定小
- D. 上述三种说法都不对

解析: 所谓算法的时间复杂度, 是指执行算法所需要的计算工作量。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。在一个算法的空间复杂度大的情况下, 它的时间复杂度只是可能很大, 具体视情况而定, 反之亦然。

答案: D

10. 在长度为 64 的有序线性表中进行顺序查找, 最坏情况下需要比较的次数为_____。(2006年9月)

- A. 63
- B. 64
- C. 6
- D. 7

解析: 在进行顺序查找过程中, 如果线性表中的第一个元素就是被查找元素, 则只需做一次比较就查找成功, 查找效率最高; 但如果被查找的元素是线性表中的最后一个元素, 或者被查找元素根本不在线性表中, 则为了查找这个元素需要与线性表中所有的元素进行比较, 这是顺序查找的最坏情况。题目中所给出的有序线性表的长度为 64, 最坏情况下要比较 64 次。

答案: B