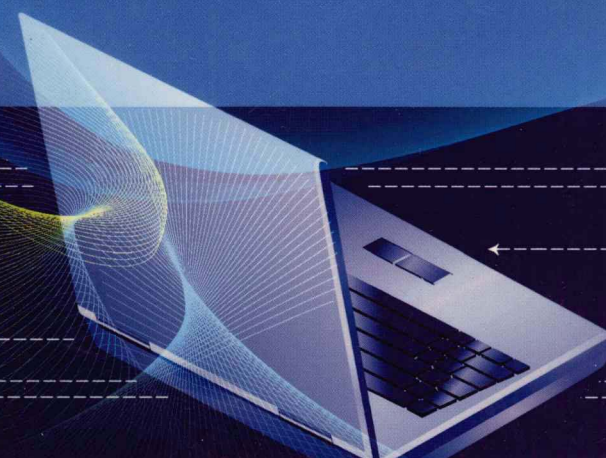




→ 数据结构

← 计算机组成原理

← 操作系统

← 计算机网络

全国研究生 计算机统一考试习题详解 (2009年新大纲)

■ 梁 旭 张振琳 黄 明 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

全国研究生计算机统一考试习题详解 (2009 年新大纲)

数据结构 计算机组成原理
操作系统 计算机网络

梁 旭 张振琳 黄 明 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是根据“全国研究生入学考试(计算机专业)考试大纲(2009年)”,精选各个著名高校历年考研真题和习题编写的。本书主要包括5大部分:第1部分是数据结构;第2部分是计算机组成原理;第3部分是操作系统;第4部分是计算机网络;第5部分是模拟试卷及参考答案;另外,还包括应试策略和附录。

本书紧扣考试大纲,广泛收集了近几年全国20余所重点高校考研试卷,对典型的真题进行了深入、细致的分析和解答。叙述通俗易懂,每门课程都附有习题及参考答案。书中还给出了模拟试卷,可用于考前训练。

本书适合报考计算机专业研究生的考生有针对性地进行专业课的复习,也适合希望深入学习计算机专业知识的高校学生作为学习相关课程的参考书,同时也适用于讲授该课程的教师及自修该课程的其他人员。

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

全国研究生计算机统一考试习题详解:2009年新大纲/梁旭编著. —北京:电子工业出版社,2008.10
ISBN 978-7-121-07479-0

I. 全… II. 梁… III. 电子计算机—研究生—入学考试—解题 IV. TP3-44

中国版本图书馆CIP数据核字(2008)第152383号

责任编辑:赵平

印刷:北京天宇星印刷厂

装订:涿州市桃园装订有限公司

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:850×1168 1/16 印张:24.75 字数:819千字

印次:2008年10月第1次印刷

印数:5000册 定价:39.00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

一年一度的研究生入学考试即将开始，这也意味着新一轮复习的开始。如今研究生入学考试已经成为继高考之后全国第二大考试，而随着 IT 产业的高速发展，计算机专业几乎成为全国最热门的专业之一。

2009 年，全国硕士研究生入学统一考试新大纲对计算机科学与技术专业的研究生入学考试专业课进行了调整，将以往专业课由各个学校独立命题，调整为全国统一命题形式。计算机科学与技术学科的初试科目调整后为 4 门，即政治理论、外语、数学一和计算机学科专业基础综合，卷面满分值分别为 100 分、100 分、150 分和 150 分。计算机学科专业基础综合的考试内容包括：数据结构、计算机组成原理、操作系统和计算机网络，重点考查考生掌握相关基础知识、基本理论和分析与解决问题的能力。

本书针对全国统一命题要求，紧扣新大纲，精选各个著名高校历年考研真题，进行详尽的解析，并给出答案，使读者可以真正掌握题目的解题思路和方法。其内容覆盖了计算机学科专业基础综合所涉及的 4 大部分。既强化了重点，又进一步细化了知识点，以帮助读者了解新大纲的要点，提高复习效率。

本书具有以下两个特点：

- (1) 紧密结合新大纲；
- (2) 全面而有重点地选择真题。

本书紧扣国家教育部制定的研究生入学考试大纲，合理安排书中各个章节内容，条理清晰，详略分明，深入浅出，解疑答惑，并广泛收集近年全国 20 余所重点高校考研试卷，对典型的真题进行了深入剖析，以求达到举一反三的目的，使读者能掌握这四门专业课程的主要原理和方法。

对于专业课的复习，必须注重“全面”与“重点”相结合，因为一门课程的内容非常多，考试只是考到其中的一小部分，所谓重点就是出题概率较高，并且所占分值也较高的内容。编者通过仔细研究各个学校的历年专业课试题，发现其中一些内容重复性很强，对这些相关典型题目进行了细致、深入的分析和解答，并进行了适当的扩展，使读者能够明确整个题目的分析过程和需要注意的地方，而不仅是知道答案而已，从而达到融会贯通的境界。对于每章的例题，读者应注重其解题的思路，而不应拘泥于特定的题目，为了使读者有更多的练习机会，每门课程都配有大量的习题及参考答案。最后还附有两套模拟试卷，以配合考生的学习。

本书由梁旭、张振琳、黄明共同编写，同时要特别感谢张韶华、孟旭升、唐玲、牛文颖、刘剑等，他们做了大量的工作。由于编著者水平有限，编写时间仓促，书中错误和不妥之处在所难免，请读者和专家批评指正。

读者在使用本书的过程中如有问题，可与下列 E-mail 联系：liangxu00@263.net。

最后，预祝广大考生旗开得胜，万事如意！

编著者

2008 年 8 月

目 录

应试策略	1
第1部分 数据结构	2
考查目标和知识点解析	2
第1章 概论	4
一、单项选择题	4
二、综合应用题	6
第2章 线性表	8
一、单项选择题	8
二、综合应用题	12
第3章 栈、队列和数组	20
一、单项选择题	20
二、综合应用题	26
第4章 树与二叉树	37
一、单项选择题	37
二、综合应用题	42
第5章 图	54
一、单项选择题	54
二、综合应用题	57
第6章 查找	66
一、单项选择题	66
二、综合应用题	71
第7章 排序	77
一、单项选择题	77
二、综合应用题	81
习题	91
一、单项选择题	91
二、综合应用题	96
参考答案	98
一、单项选择题	98
二、综合应用题	98
第2部分 计算机组成原理	109
考查目标和知识点解析	109
第8章 计算机系统概述	110
一、单项选择题	110
二、综合应用题	111
第9章 数据的表示和运算	113
一、单项选择题	113

二、综合应用题	118
第 10 章 存储器层次结构	125
一、单项选择题	125
二、综合应用题	129
第 11 章 指令系统	140
一、单项选择题	140
二、综合应用题	142
第 12 章 中央处理器 (CPU)	149
一、单项选择题	149
二、综合应用题	152
第 13 章 总线	160
一、单项选择题	160
二、综合应用题	162
第 14 章 输入/输出 (I/O) 系统	163
一、单项选择题	163
二、综合应用题	168
习题	172
一、单项选择题	172
二、综合应用题	176
参考答案	179
一、单项选择题	179
二、综合应用题	179
第 3 部分 操作系统	185
考查目标和知识点解析	185
第 15 章 操作系统概述	187
一、单项选择题	187
二、综合应用题	190
第 16 章 进程管理	192
一、单项选择题	192
二、综合应用题	201
第 17 章 内存管理	228
一、单项选择题	228
二、综合应用题	234
第 18 章 文件管理	242
一、单项选择题	242
二、综合应用题	248
第 19 章 输入/输出 (I/O) 管理	257
一、单项选择题	257
二、综合应用题	258
习题	261
一、单项选择题	261
二、综合应用题	267
参考答案	276

一、单项选择题	276
二、综合应用题	276
第4部分 计算机网络	291
考查目标和知识点解析	291
第20章 计算机网络体系结构	293
一、单项选择题	293
二、综合应用题	298
第21章 物理层	301
一、单项选择题	301
二、综合应用题	304
第22章 数据链路层	309
一、单项选择题	309
二、综合应用题	320
第23章 网络层	329
一、单项选择题	329
二、综合应用题	333
第24章 传输层	341
一、单项选择题	341
二、综合应用题	342
第25章 应用层	344
一、单项选择题	344
二、综合应用题	348
习题	352
一、单项选择题	352
二、综合应用题	355
参考答案	356
一、单项选择题	356
二、综合应用题	356
第5部分 模拟试卷及参考答案	359
全真模拟试卷(一)	359
一、单项选择题	359
二、综合应用题	361
全真模拟试卷(一) 分析解答	363
一、单项选择题	363
二、综合应用题	363
全真模拟试卷(二)	368
一、单项选择题	368
二、综合应用题	370
全真模拟试卷(二) 分析解答	373
一、单项选择题	373
二、综合应用题	373
附录 2009年计算机专业硕士入学考试大纲	376
参考文献	386

应 试 策 略

数据结构、计算机组成原理、操作系统、计算机网络这四门课程都是计算机专业的基础课程，每门课程所涉及的内容十分广泛，具有知识点多、难度和跨度大等特点。而硕士考试中的计算机学科专业基础综合要同时涵盖这四门课程，其难度可想而知。

作为即将参加考研的考生，如何在有限的时间内取得更好的成绩，是每个考生都非常关心的问题。为此，我们从考生的实际出发，结合多年的教学经验，总结了一些迎考备战中应注意的问题，希望能在考生的复习过程中起到一定的指导作用。

（一）深入理解考试大纲

全国计算机硕士统一考试计算机学科专业基础综合考试大纲对考试的基本要求、考试内容都做了具体规定（详见附录部分）。

在计算机考研专业基础课统考科目中，一共考查数据结构、计算机组成原理、操作系统、计算机网络四门课程，满分为 150 分。其中数据结构 45 分、计算机组成原理 45 分、操作系统 35 分、计算机网络 25 分。统考只有两种题型：单项选择题和综合应用题，选择题占了 80 分，共四十道题，从这点上可以很明显地看出来，考查的知识点将会相对地比较全面。此外，综合应用题占了 70 分。大纲的考查范围说的比较笼统，考查的知识点方面也只是把有关教材的章节名列举出来，并没有强调每个知识点应该掌握到什么程度。因此考生在复习的过程中要全面复习，注重基础，重点是基本概念的理解。

（二）应试策略

1. 选择合适的教材。近年来，这四门课程的书籍越来越丰富。国内许多高校都有各自不同的自编教材，同时，一些国外的经典教材也陆续被引进。这些教材，在讲授同一门课程时，侧重点往往有所不同，覆盖的知识面及知识点的取舍不尽相同，对一些概念的叙述和解释往往也存在着一定的差异。这些问题考生在复习时都应该加以注意。

2. 重视基础题的练习。这四门课程都有许多基本概念，这些题目看似简单，但很容易出错，而且这也反映出考生对基础概念的理解是否透彻，尤其在一些难度较大的题目中，这些基础概念也是解题的基础。

第1部分 数据结构

考查目标和知识点解析

【考查目标】

1. 理解数据结构的基本概念, 掌握数据的逻辑结构、存储结构及其差异, 以及各种基本操作的实现。
2. 在掌握基本的数据处理原理和方法的基础上, 能够对算法进行设计与分析。
3. 能够选择合适的数据结构和方法进行问题求解。

【知识点解析】

1. 线性表

线性表是一种最简单的数据结构, 但线性表在线性结构的学习乃至整个数据结构学科的学习中都具有非常重要的作用。其中系统地引入链式存储的概念, 链式存储概念是整个数据结构学科的重中之重, 无论哪一章都涉及了这个概念, 所以需要透彻理解。

这部分要识记线性表相关的基本概念和线性表的结构特点, 掌握线性表的顺序存储方式和链式存储方式的实现方法及相应的运算, 掌握链表与顺序表的相似及不同之处。掌握几种常用的链表的特点和运算。在理解线性表的顺序存储及链式存储的情况下, 其不同的优缺点的比较及各自适用的场合。另外, 还要掌握线性表的基本应用。

2. 栈、队列和数组

栈、队列和数组都属于线性结构的拓展, 栈和队列是特殊的线性表, 是操作受限的线性表, 数组是数据元素为是非原子类型的线性表。这部分要识记栈、队列的定义及相关数据结构的概念, 理解栈和队列的特点与运算的实现, 掌握栈和递归的关系, 掌握循环队列的各种判断和运算, 了解栈和队列的应用。

对于数组这部分, 主要掌握数组的存储结构, 例如按行优先、按列优先等, 掌握多维数组中某个元素的位置求解。了解特殊矩阵的定义, 掌握特殊矩阵的压缩存储。

3. 树与二叉树

树和二叉树是考试的重点和难点章节, 二叉树和树是两种不同的概念。在这个部分, 要掌握树的定义、二叉树的定义及主要特征(特殊的二叉树、二叉树的性质)。掌握二叉树的性质和不同存储方式及各自的优缺点和适用场合。熟练掌握二叉树的遍历方法。

在特殊的二叉树中, 完全二叉树的概念是必须要熟练掌握的。另外, 要掌握线索二叉树的基本概念和构造, 二叉排序树、平衡二叉树的基本概念和应用。

要掌握树的存储结构和遍历、森林的遍历、树和二叉树的转换、森林和二叉树的转换等知识。最后就是树的应用, 要掌握哈夫曼树的构造和等价类问题。

4. 图

在数据结构中, 图的结构是最复杂的, 这里的概念也是最多的。需要掌握图的基本概念(有向图、无向图、连通、路径、子图、出度、入度、生成树、最短路径、关键路径等)。掌握相关概念的计算方法。

图的存储主要有邻接矩阵法和邻接表法, 要掌握有向图和无向图的这两种存储方法, 要掌握图的连通和

存储方法之间的关系以及在图的两种存储方法中识别图的各种基本信息。

图的遍历方法有深度优先搜索和广度优先搜索，要熟练掌握图的两种遍历算法。

图的基本应用是很多的，也比较复杂，要掌握这些应用的复杂度分析。掌握最小生成树的概念及构造，掌握拓扑排序、关键路径的定义与计算方法以及最短路径的计算。

5. 查找

查找是重点和难点，概念较多，相互联系较为紧密。查找的基本方法主要有顺序查找法、折半查找法、B-树、散列（Hash）表及其查找。需要识记基本概念，掌握平均查找长度的计算，掌握线性表上的查找，熟练掌握树表上的查找。熟练掌握哈希表的基本思想和方法。

另外，要掌握各种查找算法的分析及应用，掌握各种查找在查找成功、查找失败的情况下的最好、平均、最坏的平均查找次数的计算方法。

6. 内部排序

内部排序也是重点和难点的章节，且概念更多，联系更为紧密，概念之间容易混淆。在这一部分中，主要需掌握直接插入排序、折半插入排序、冒泡排序（bubble sort）、简单选择排序、希尔排序、快速排序、堆排序、二路归并排序、基数排序的基本概念和方法。掌握这些排序方法的思想 and 流程，以及它们之间的区别。

在这部分，一个很重要的考查点就是各种内部排序算法的比较，考查各种排序算法及其思想以及优缺点和性能指标（时间复杂度，空间复杂度），进而了解各种排序方法适用的场合。

最后，就是要掌握内部排序算法的基本应用以及算法的实现。

第1章 概 论

一、单项选择题

【例 1-1】对数据结构，下列结论中不正确的是（ ）。(华南理工大学 2005 年试题)

- A. 相同的逻辑结构，对应的存储结构也必相同
- B. 数据结构由逻辑结构、存储结构和基本操作三个方面组成
- C. 数据存储结构就是数据逻辑结构的机内的实现
- D. 对数据基本操作的实现与存储结构有关

解析：数据结构由数据的逻辑结构、数据的存储结构和基本操作三个方面组成，所以选项 B 的结论是正确的。

数据的逻辑结构可以看作是从具体问题抽象出来的数学模型，它与数据的存储无关。我们研究数据结构的目的是为了在计算机中实现对它的操作，为此还需要研究如何在计算机中表示一个数据结构。数据结构在计算机中的标识（又称映像）称为数据的存储结构，它所研究的是数据结构在计算机中的实现方法，包括数据结构中元素的表示及元素间关系的表示，所以选项 C 的结论是正确的。

同一种逻辑结构可以采用不同的存储方法，从而得到不同的存储结构，所以选项 A 的结论是不正确的。

数据的运算是定义在数据的逻辑结构上的，常用的运算包括：检索、插入、删除、排序等，这些操作的实现要依赖于数据的存储结构，所以选项 D 的结论是正确的。故答案是 A。

答案：A

【例 1-2】在数据结构中，从逻辑上可以将之分为（ ）。(中南大学 2005 年试题)

- A. 动态结构和静态结构
- B. 紧凑结构和非紧凑结构
- C. 内部结构和外部结构
- D. 线性结构和非线性结构

解析：数据的逻辑结构是从逻辑关系上描述数据，它与数据的存储无关，是独立于计算机的。数据的逻辑结构有两大类：线性结构和非线性结构。线性结构的逻辑特征是：有且仅有一个开始结点和一个终端结点，并且所有的结点最多只有一个直接前驱和直接后继。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。所以选项 D 是正确的。

答案：D

【例 1-3】下面说法错误的是（ ）。

- (1) 算法原地工作的含义是指不需要任何额外的辅助空间
 - (2) 在相同的规模 n 下，复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2n)$ 的算法
 - (3) 所谓时间复杂度是指最坏情况下，估算算法执行时间的一个上界
 - (4) 同一个算法，实现语言的级别越高，执行效率就越低
- A. (1) B. (1), (2) C. (1), (4) D. (3)

解析：下面对各个选项分别分析：算法的空间复杂度是算法所需存储空间的量度。其中所考虑的算法所需存储空间是除输入数据和算法本身之外的辅助空间，若所需辅助空间相对于输入量来说是常数，则称此算法是原地工作的。故说法 (1) 是不正确的。

对于算法时间复杂度的计算，通常是从算法中选取一种对于所研究的问题来说是基本操作的原操作，以该基本操作在算法中重复执行的次数作为算法时间复杂度的依据。这种衡量效率的办法所得出的不是时间量，而是一种增长趋势的量度。所以在相同的规模 n 下，复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2n)$ 的算法，故说法 (2) 是正确的。

很多算法的时间复杂度不仅与问题的规模有关,而且还与它所处理的数据集的状态有关。通常是根据数据集中可能出现的最坏情况估计出算法的最坏时间复杂度。故说法(3)是正确的。

程序在计算机上运行时所消耗的时间取决于许多因素,其中之一是书写程序的语言,对于同一个算法,实现语言的级别越高,执行效率越低。故说法(4)是正确的。所以答案是A。

答案: A

【例 1-4】算法的时间复杂度取决于()。

A. 问题的规模

B. 待处理数据的初态

C. A 和 B

解析: 算法的运行效率包括时间效率和空间效率。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,以该原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 $T(n)$,也就是说算法的时间复杂度取决于问题的规模。

同时,很多算法的时间复杂度不仅是问题规模的函数,还与它所处理的数据集的状态有关,在这种情况下,通常是根据数据集中可能出现的最坏情况,估计出算法的最坏时间复杂度,根据数据集中可能出现的最好情况,估计出算法的最好时间复杂度,有时也对数据集的分布做出某种假定,并讨论算法的平均时间复杂度,所以算法的时间复杂度也取决于待处理数据的初态。故选项C是正确的。

答案: C

【例 1-5】在下面的程序段中,对 x 的赋值语句的频度为()。

```
FOR i:=1 TO n DO
  FOR j:=1 TO n DO
    x:=x+1;
```

A. $O(2n)$

B. $O(n)$

C. $O(n^2)$

D. $O(\log_2 n)$

解析: 一般情况下,这类题目中,求解频度的语句都是在多重循环中,对应的计算方法是:首先查看各个循环的循环变量的取值范围,然后根据取值范围列出频度公式即可。

$$T(n) = \sum_{i=1}^n \sum_{j=1}^n 1 = \sum_{i=1}^n n = n \times n = n^2$$

在公式的计算过程中按照从内层向外层逐层计算。

本题循环变量范围是常量,可以不列出公式而直接求出,但是许多考题中循环变量范围是变量,只有掌握公式的方式才能以不变应万变。

答案: C

【例 1-6】下面程序段的时间复杂性为()。

```
y=0;
while(n>=(y+1)*(y+1))
{
  y++;
}
```

A. $O(n)$

B. $O(n^2)$

C. $O(\sqrt{n})$

D. $O(1)$

解析: 求解时间复杂度的一般方法是:找出所有语句中执行频度最大的那条语句,然后取其数量级并放入 O 记法中即可。设 while 语句的执行频度为 $T(n)$,显然有:

$T(n) * T(n) \leq n$, 即 $T(n)^2 \leq n$, 所以 $T(n) \leq \sqrt{n} = O(\sqrt{n})$ 。所以答案是C。

答案: C

【例 1-7】计算机执行下面的语句时,语句 s 的执行次数为()。

```
FOR(i=1; i<n-1; i++)
```

```
FOR(j=n;j>=i;j--)
```

```
s;
```

A. $n(n+1)/2$ B. n^2 C. $(n+3)(n-2)/2$ D. $n(n-2)/2$

解析: 本题中频度最大的语句是 s, 该语句的频度 $T(n)$ 为:

$$T(n) = \sum_{i=1}^{n-2} \sum_{j=i}^n 1 = \sum_{i=1}^{n-2} (n-i+1) = n \times (n-2) - \sum_{i=1}^{n-2} i + (n-2) = \frac{(n+3)(n-2)}{2}$$

所以答案是 C。

答案: C

【例 1-8】在下面的程序段中, 对 x 的赋值语句的频度为 () (表示为 n 的函数)。

```
FOR i:=1 TO n DO
```

```
FOR j:=1 TO i DO
```

```
FOR k:=1 TO j DO
```

```
x:=x+delta;
```

A. $n(n+1)^2/2$ B. n^3 C. $n(n+1)(n+2)/6$ D. $n^2(n-2)/6$

解析: 本题中对 x 赋值的语句 $x := x + \text{delta}$ 是频度最大的, 观察各层循环可以发现, 内层循环的执行次数虽然与问题规模 n 没有直接关系, 但却与外层循环的变量取值有关, 而最外层循环的次数直接与 n 有关, 因此从内向外逐层计算该语句的执行次数:

$$T(n) = \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^j 1 = \sum_{i=1}^n \sum_{j=1}^i j = \sum_{i=1}^n \frac{i(i+1)}{2} = \frac{\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2}}{2} = \frac{n(n+1)(n+2)}{6}$$

所以答案是 C。

答案: C

二、综合应用题

【例 1-9】有以下算法, 分析其时间复杂度。(华南理工大学 2005 年试题)

```
i=1;
```

```
while(i*i*i<=n) i++;
```

解析: 求解时间复杂度的一般方法是: 找出所有语句中执行频度最大的那条语句, 然后取其数量级并放入 O 记法中即可。一般情况下, 对循环语句中只需考虑循环体中语句的执行次数, 本题中频度最大的语句是 $i++$, 由循环条件 $i*i*i \leq n$, 可知当其 $i \leq \log_3 n$ 时, 执行循环体, 循环体的执行频度是 $\log_3 n$, 所以该程序段的时间复杂度为 $O(\log_3 n)$ 。

答案: 程序段中频度最大的语句是 $i++$, 由循环条件 $i*i*i \leq n$, 可知当 $i \leq \log_3 n$ 时, 执行循环体, 循环体的执行频度是 $\log_3 n$, 所以该程序段的时间复杂度为 $O(\log_3 n)$ 。

【例 1-10】设 n 是偶数, 试计算运行下列程序段后 m 的值并给出该程序段的时间复杂度。

```
m:=0;
```

```
FOR i:=1 TO n DO
```

```
FOR j:=2*i TO n DO
```

```
m:=m+1;
```

解析: 本题中频度最大的语句是 $m:=m+1$, 由于内循环从 $2*i$ 到 n , 即 i 的最大值满足 $2i \leq n$, 所以 $i \leq n/2$, 该语句的频度为:

$$T(n) = \sum_{i=1}^{n/2} \sum_{j=2i}^n 1 = \sum_{i=1}^{n/2} (n-2i+1) = n \times \frac{n}{2} - 2 \sum_{i=1}^{n/2} i + \frac{n}{2} = \frac{n^2}{4} = O(n^2)$$

该段程序段的时间复杂度为 $O(n^2)$ 。

题目中要求出程序运行后 m 的值，由程序可知，循环前， m 的初始值为 0，在双循环体中， $m:=m+1$ ，所以 m 的值就是语句 $m:=m+1$ 的执行次数，即 m 的值为 $\frac{n^2}{4}$ 。

答案：算法中频度最大的语句是 $m:=m+1$ ，由于内循环从 $2*i$ 到 n ，即 i 的最大值满足 $2i \leq n$ ，所以 $i \leq n/2$ ，该语句的频度为：

$$T(n) = \sum_{i=1}^{n/2} \sum_{j=2i}^n 1 = \sum_{i=1}^{n/2} (n-2i+1) = n \times \frac{n}{2} - 2 \sum_{i=1}^{n/2} i + \frac{n}{2} = \frac{n^2}{4} = O(n^2)$$

该程序段的时间复杂度为 $O(n^2)$ ， m 的值为 $\frac{n^2}{4}$ 。

第2章 线性表

一、单项选择题

【例 2-1】对于顺序存储的线性表，访问结点和增加结点的时间复杂度为（ ）。(中国科学院研究生院 2007 年试题)

A. $O(n) O(n)$ B. $O(n) O(1)$ C. $O(1) O(n)$ D. $O(1) O(1)$

解析：线性表的顺序存储是指在内存中用地址连续的一块存储空间顺序存放线性表的各元素，用这种存储形式存储的线性表称其为顺序表。因为内存中的地址空间是线性的，因此，用物理上的相邻实现数据元素之间的逻辑相邻关系是既简单又自然的。设 a_1 的存储地址为 $\text{Loc}(a_1)$ ，每个数据元素占 d 个存储地址，则第 i 个数据元素的地址为：

$$\text{Loc}(a_i) = \text{Loc}(a_1) + (i-1) * d \quad 1 \leq i \leq n$$

这就是说，只要知道顺序表首地址和每个数据元素所占地址单元的个数就可求出第 i 个数据元素的地址，这也是顺序表具有按数据元素的序号随机存取的特点，所以顺序表中访问结点的时间复杂度为 $O(1)$ 。

顺序表上的增加结点运算，时间主要消耗在数据的移动上，在第 i 个位置上插入 x ，从 a_i 到 a_n 都要向下移动一个位置，共需要移动 $n-i+1$ 个元素，而 i 的取值范围为： $1 \leq i \leq n+1$ ，即有 $n+1$ 个位置可以插入。设在第 i 个位置上作插入的概率为 P_i ，则平均移动数据元素的次数：

$$E_{in} = \sum_{i=1}^{n+1} p_i (n-i+1)$$

设： $P_i = 1/(n+1)$ ，即为等概率情况，则：

$$E_{in} = \sum_{i=1}^{n+1} p_i (n-i+1) = \frac{1}{n+1} \sum_{i=1}^{n+1} (n-i+1) = \frac{n}{2}$$

这说明：在顺序表上做插入操作需移动表中一半的数据元素。显然时间复杂度为 $O(n)$ 。

故答案是 C。

答案：C

【例 2-2】单链表中空指针数为 0 时，该链表为（ ）。(电子科技大学 2006 年试题)

A. 双向链表

B. 广义表

C. 循环链表

D. 哈希表

解析：对于单链表而言，最后一个结点的指针域是空指针。双向链表是指链表中每个结点有两个指针域，一个指向其后继，另一个指向其前驱，其中链表表尾结点的后继指针为空。而循环链表是将链表头尾结点相连构成的，链表表尾结点的指针指向链表头指针，即每个指针域都非空。所以循环链表中没有空的指针域。故答案是 C。

答案：C

【例 2-3】对于一个头指针为 head 的带头结点单链表，判断表为空的条件是（ ）。(中国科学院研究生院 2007 年试题)

A. $\text{head} = \text{NULL}$ B. $\text{head} \uparrow .\text{next} = \text{NULL}$ C. $\text{head} \uparrow .\text{next} = \text{head}$ D. $\text{head} \neq \text{NULL}$

解析：为了方便操作，在单链表的头部加入一个“头结点”形成带头结点单链表，头结点的类型与数据结点一致，标识链表的头指针变量中存放该结点的地址，头结点的加入完全是为了运算的方便，它的数据域无定义，指针域中存放的是第一个数据结点的地址，空表时为 NULL。

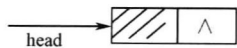


图 2-1 带头结点的空单链表

图 2-1 是带头结点的单链表为空表的示意图。

所以当链表为空时， $\text{head} \uparrow .\text{next} = \text{NULL}$

故答案是 B。

答案: B

【例 2-4】线性表若采用链表存储结构, 要求内存中可用存储单元地址 ()。(湖南大学 2006 年试题)

- A. 必须连续 B. 部分地址必须连续 C. 一定不连续 D. 连续不连续均可

解析: 链表是通过一组任意的存储单元来存储线性表中的数据元素的, 为建立起数据元素之间的线性关系, 对每个数据元素 a_i , 除了存放数据元素自身的信息 a_i 之外, 还需要和 a_i 一起存放其后继 a_{i+1} 所在的存储单元的地址, 这两部分信息组成一个“结点”, 其中存放数据元素信息的称为数据域, 存放其后继地址的称为指针域。也就是说在链表中数据元素 a_i 和 a_{i+1} 之间的逻辑关系是靠指针域来维持的, 而与 a_i 和 a_{i+1} 的地址位置无关, 所以说链表和顺序表不同, 其存储单元既可以是连续的, 也可以是不连续的, 甚至是零散的分布在内存中的任何位置上。所以选项 D 是正确的。

答案: D

【例 2-5】若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算, 则利用 () 存储方式最节省时间。

- A. 顺序表 B. 双链表 C. 带头结点的双循环链表 D. 单循环链表

解析: 本题考查线性表的各种存储结构的优缺点, 这类题目在考试中经常出现, 通常的题目都是给出线性表的常用操作, 来确定最有效的存储方式。

线性表的顺序存储具有按元素序号随机访问的特点, 但它的缺点是做插入删除操作时, 平均移动表中一半的元素, 当数据元素的信息量较大且表较长时, 效率较低。链表的优缺点恰好与顺序表相反。所以如果经常做的运算是按序号访问数据元素, 则顺序表优于链表; 而如果经常做的运算是频繁地插入删除, 则宜选择链式存储。

对本题来说, 如果没有认真分析题目, 只是看到“进行插入和删除运算”, 则很容易选择链表结构, 分析一下, 题目中“存取任一指定序号的元素”操作就是进行随机访问, 这是顺序存储的优势, 而且这里的插入删除不是在任意位置的, 而是线性表最后位置进行, 所以这种插入删除不需要移动元素, 时间性能为 $O(1)$ 。所以这种情况下, 采用顺序表存储方式最节省时间。故答案是 A。

答案: A

【例 2-6】在非空双向循环链表中由 q 所指的那个链接点前插入一个 p 指的链接点的动作对应的语句依次为 $\text{rlink}(p) \leftarrow q, \text{llink}(p) \leftarrow \text{llink}(q), \text{llink}(q) \leftarrow p, ()$ 。(空白处为一条赋值语句)

- A. $\text{rlink}(q) \leftarrow p$ B. $\text{rlink}(\text{llink}(q)) \leftarrow p$ C. $\text{rlink}(\text{llink}(p)) \leftarrow p$ D. $\text{rlink}(\text{rlink}(p)) \leftarrow p$

解析: 在单链表的每个结点里再增加一个指向前驱的指针域 prior , 结点的结构如图 2-2 所示, 这样形成的链表中有两条方向不同的链, 用这种结点组成的链表称为双向链表。

双向链表中结点的插入: 设 q 指向双向链表中某结点, p 指向待插入的值为 x 的新结点, 将 $*p$ 插入到 $*q$ 的前面, 插入示意图如图 2-3 所示。

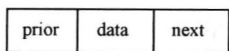


图 2-2 双向链表示意图

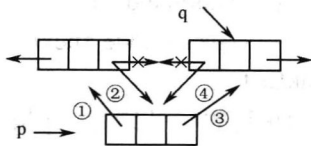


图 2-3 双向链表中的结点插入

操作如下:

- ① $p \rightarrow \text{prior} = q \rightarrow \text{prior};$
- ② $q \rightarrow \text{prior} \rightarrow \text{next} = p;$
- ③ $p \rightarrow \text{next} = q;$
- ④ $q \rightarrow \text{prior} = p;$

指针操作的顺序不是唯一的，但也不是任意的，操作①必须放到操作④的前面完成，否则*q的前驱结点的指针就丢掉了。

本题目中对于双链表的指针域采用不同的表示方法，可以知道， $\text{llink}(p)$ 表示p的前驱，即 $p \rightarrow \text{prior}$ ， $\text{rlink}(p)$ 表示p的后继，即 $p \rightarrow \text{next}$ ，题目中语句的对应关系为：

- ③ $\text{rlink}(p) \leftarrow q$
- ① $\text{llink}(p) \leftarrow \text{llink}(q)$
- ④ $\text{llink}(q) \leftarrow p$

所以空白处应该是操作②，即 $\text{rlink}(\text{llink}(p)) \leftarrow p$ 。所以选项C是正确的。这里要特别强调一下，很容易错误选择B，即 $\text{rlink}(\text{llink}(q)) \leftarrow p$ ，其实如果简单地把 $q \rightarrow \text{prior} \rightarrow \text{next} = p$ 转换表示方法得到的就是 $\text{rlink}(\text{llink}(q)) \leftarrow p$ 。但这是错误的，原因在于指针的不同执行顺序，按题目的执行顺序，在前三步操作后， $\text{llink}(q)$ 不再是指向q之前的前驱，而是指向了p，所以 $\text{rlink}(\text{llink}(q))$ 指向的仍是q本身。

所以在双链表的题目中，要特别注意指针的操作顺序，注意前面的操作对后面操作的影响。

答案：C

【例 2-7】某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用（ ）存储方式最节省运算时间。

- A. 单链表
- B. 仅有头指针的单循环链表
- C. 双链表
- D. 仅有尾指针的单循环链表

解析：本题主要考查各种链表的优缺点。对于单链表而言，最后一个结点的指针域是空指针，如果将该链表头指针置入该指针域，则使得链表头尾结点相连，就构成了单循环链表。单循环链表的主要优点是：从表中任意结点出发都能通过后移操作而遍历整个循环链表。在单循环链表中，如用头指针标识，找到链表的尾结点，其时间复杂性为 $O(n)$ ，如果采用尾指针标识，无论找头结点、首结点还是尾结点都很方便，它们的存储位置分别是 $\text{rear} \rightarrow \text{next}$ 、 $\text{rear} \rightarrow \text{next} \rightarrow \text{next}$ 和 rear ，其时间性能为 $O(1)$ 。

双链表是一种对称结构，每个结点既有指向后继的指针域，又有指向前驱的指针域，这就使得插入操作及删除操作都非常方便。但是这种存储方式需要增加空间的代价。

本题中，最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，从上面的分析知道，采用有尾指针的单循环链表查找最后一个元素和第一个元素的时间都是 $O(1)$ ，所以答案是D。

答案：D

【例 2-8】在非空线性链表中由p所指的链接点后面插入一个由q所指的链接点的过程是依次执行动作（ ）。

- A. $\text{link}(q) \leftarrow p; \text{link}(p) \leftarrow q;$
- B. $\text{link}(q) \leftarrow \text{link}(p); \text{link}(p) \leftarrow q;$
- C. $\text{link}(q) \leftarrow \text{link}(p); p \leftarrow q;$
- D. $\text{link}(p) \leftarrow q; \text{link}(q) \leftarrow p;$

解析：链表是由一个个结点构成的，结点结构如图2-4所示，定义如下：

```
typedef struct node
{
    datatype data;
    struct node *next;
} LNode, *LinkList;
```

后插结点：设p指向单链表中某结点，q指向待插入的值为x的新结点，将*q插入到*p的后面，插入示意图如图2-5所示。

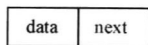


图 2-4 单链表示意图

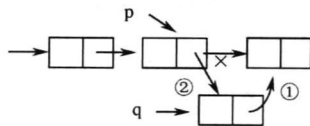


图 2-5 在 *p 之后插入 *q