



王道考研系列

—www.cskaoyan.com—

2014年 计算机专业基础综合考试 真题思路分析

王道论坛由国内一流名校计算机研究生共同创办，致力于给报考计算机专业研究生的考生提供帮助和指导。王道论坛是唯一一家专注于计算机学生考研和就业的社区。

“王道考研系列”融入了论坛众多名校高分选手的智慧，以及论坛交流精华，我们希望将其打造成“书本+在线”的学习方式，对于书中的疑难点，欢迎在论坛交流讨论。

“王道集训营”是王道创始团队结合王道牛人、哈工大校友和腾讯/百度等名企资源，给面临就业的学生或想提升编程和算法能力的研究生提供的线下集中式辅导。

王道论坛 组编

2014年
计算机专业基础综合考试
真题思路分析

更多计算机
考研、学习交流
www.cskaoyan.com



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

TP3/937

2013

王道考研系列

2014 年计算机专业基础综合考试 真题思路分析

王道论坛 组编

RFID

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

对于任何考试,真题永远都是最宝贵的资源,研究生入学考试更是如此。历年真题不仅需要实战模拟,更需要吃透,而由真题引申出的周边核心考点,也一定要牢固掌握。本书在认真、细致剖析 2009—2013 年计算机专业硕士研究生入学考试统考真题的基础上,分数据结构、计算机组成原理、操作系统、计算机网络四篇,对真题所涉及的核心知识点进行了详尽的分析,给出了真题的分布、命题重点,以及需要重点掌握的考点。

本书可作为考生参加计算机专业研究生入学考试的复习用书,也可作为计算机专业学生的辅导用书。

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

2014 年计算机专业基础综合考试真题思路分析/王道论坛组编. —北京:电子工业出版社,2013.9
(王道考研系列)

ISBN 978-7-121-21571-1

I. ①2… II. ①王… III. ①电子计算机—研究生—入学考试—题解 IV. ①TP3-44

中国版本图书馆 CIP 数据核字(2013)第 229459 号

策划编辑:谭海平

责任编辑:郝黎明

印 刷:涿州市京南印刷厂

装 订:涿州市京南印刷厂

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:13.75 字数:352 千字

印 次:2013 年 9 月第 1 次印刷

定 价:36.00 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。

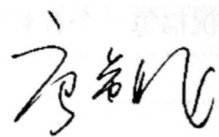
序 言

当前，随着我国经济和科技高速发展，特别是计算机科学突飞猛进的发展，对计算机相关人才，尤其是中高端人才的需求也将不断增长。硕士研究生入学考试可视为人生的第二次大考试，它是改变命运、实现自我理想的又一次机会，而计算机专业一直是高校考研的热门专业之一。

自计算机专业研究生入学考试实行统一命题以来，初试科目包含了最重要的四门基础课程（数据结构、计算机组成原理、操作系统、计算机网络），很多学生普遍反映找不到方向，复习也无从下手。倘若有一本能够指导考生如何复习的好书，必将对考生的帮助匪浅。我的学生风华他们策划和编写了这一系列的计算机专业考研辅导书，重点突出，层次分明。他们结合了自身的复习经验、理解深度以及对大纲把握程度的体会，对考生而言是很有启发和指导意义的。

计算机这门学科，任何机械式的死记硬背都是收效甚微的。在全面深入复习之后，首先对诸多知识点分清主次，并结合做题，灵活运用所掌握的知识点，再选择一些高质量的模拟试题来检测自己理解和掌握的程度，查漏补缺。这符合我执教 40 余年来一直坚持“教材—习题集—试题库”的教学体系。

从风华他们策划并组建编写团队到初稿成形，直至最后定稿，我能体会到风华和他的团队确实倾注了大量的精力。这套书的出版一定会受到广大考研学生的欢迎，它会使你在考研的路上得到强有力的帮助。



2013 年 8 月

前言

最后 50 天，拼的是心态与意志！

你的身边会陆续有一些意志不太坚定的同学放弃自己的梦想，他们的言行或许会影响到你的心情与心态。但我相信，一路用心坚持到最后的人，一定会收获属于自己的那份回报。成功并非遥不可及，坚定自己的梦想，做最好的自己！

对于任何一门考试，真题永远都是最宝贵的资源，研究生入学考试更是如此。真题不仅仅需要实战模拟，更需要方方面面地吃透、再吃透。而由真题引申出的周边核心考点，也一定要牢固掌握，以前考过的知识点极有可能再次涉及，重要的考点往往容易反复出现。《2014 年计算机专业基础综合考试真题思路分析》就是一本帮助你实现这样目的的一本书！

我们在认真、细致剖析 2009—2013 年计算机统考真题的基础上，对真题所涉及的核心知识点进行了详尽分析，并在总结统考命题规律的基础上，针对命题规律编写了高质量的模拟题（即《王道 2014 年计算机专业基础综合考试最后 8 套模拟题》），这样的规律可以指明考生在最后阶段的复习方向。也正是基于我们对命题规律的把握，王道论坛组编的最后 8 套模拟题，不仅连续命中 2012—2013 年算法设计题，更是点中多道选择题和综合题的核心考点与相关思路。

在考研前的最后 30 天，同学们一定要抓住专业课复习的四大重点：①回顾以前复习中标注的重点和难点，毕竟知识都是容易遗忘的；②查漏补缺，特别是之前做错的题目，这是为了以后遇到类似的题不再出错；③仔细分析历年真题，真题涉及的知识点一定是核心考点；④做几套高质量的模拟题。

我们都是 CS 的“屌丝”，唯有自强不息，才能实现“屌丝”的梦想！

祝福每一个有梦的“屌丝”！

风华漫舞

2013 年 9 月

目 录

第 1 篇 数据结构

第 1 章 绪论	2
真题分布及命题重点	2
考点 1: 时间复杂度与空间复杂度	2
第 2 章 线性表	5
真题分布及命题重点	5
考点 2: 线性表的顺序表示	5
考点 3: 线性表的链式表示	8
第 3 章 栈和队列	13
真题分布及命题重点	13
考点 4: 栈和队列的基本性质	13
考点 5: 栈和队列的存储结构	14
考点 6: 双端队列	15
考点 7: 栈与队列的应用	16
考点 8: 特殊矩阵的压缩存储	19
第 4 章 树与二叉树	21
真题分布及命题重点	21
考点 9: 树的基本性质	21
考点 10: 二叉树的定义与性质	22
考点 11: 二叉树的遍历	23
考点 12: 二叉排序树	25
考点 13: 平衡二叉树	27
考点 14: 树、森林与二叉树的转换	31
考点 15: 线索二叉树的基本概念和构造	32
考点 16: 哈夫曼树与哈夫曼编码	34
第 5 章 图	36
真题分布及命题重点	36
考点 17: 图的基本概念	36
考点 18: 图的存储及基本操作	38
考点 19: 图的遍历	39
考点 20: 最小(代价)生成树	41
考点 21: 最短路径	42
考点 22: 拓扑排序	45



考点 23: 关键路径	46
第 6 章 查找	49
真题分布及命题重点	49
考点 24: 顺序查找法&折半查找法	49
考点 25: B 树及其基本操作、B+树的基本概念	52
考点 26: 散列 (HASH) 表	55
第 7 章 排序	58
真题分布及命题重点	58
考点 27: 插入排序	58
考点 28: 交换排序	59
考点 29: 选择排序	61
考点 30: 二路归并排序	63
考点 31: 基数排序	65
考点 32: 各种内部排序方法的比较	66
考点 33: 外部排序的思想	67
第 2 篇 计算机组成原理	
第 1 章 计算机系统概论	70
真题分布及命题重点	70
考点 34: 计算机系统层次结构	70
考点 35: 计算机的性能指标	71
第 2 章 数据的表示和运算	73
真题分布及命题重点	73
考点 36: 海明码	73
考点 37: 数据的存储方式	74
考点 38: 定点数的表示与运算	76
考点 39: IEEE754 标准	80
考点 40: 各种精度数据的转换	82
考点 41: 浮点数运算	83
第 3 章 存储系统的层次结构	85
真题分布及命题重点	85
考点 42: 半导体随机存取存储器	85
考点 43: 主存储器与 CPU 的连接	87
考点 44: 低位交叉存储器	88
考点 45: 高速缓冲存储器 (CACHE)	89
考点 46: 虚拟存储器	93
第 4 章 指令系统	97
真题分布及命题重点	97
考点 47: 指令格式	97

考点 48: 指令的寻址方式	98
考点 49: CISC 与 RISC	102
第 5 章 中央处理器	104
真题分布及命题重点	104
考点 50: CPU 的功能和基本结构	104
考点 51: 指令执行过程	105
考点 52: 数据通路的功能和基本结构	106
考点 53: 控制器的功能和工作原理	108
考点 54: 指令流水线	110
第 6 章 总线	114
真题分布及命题重点	114
考点 55: 总线分类	114
考点 56: 总线的性能指标	115
考点 57: 总线标准	116
第 7 章 输入/输出系统	117
真题分布及命题重点	117
考点 58: 外部设备	117
考点 59: 磁盘与 RAID	118
考点 60: I/O 接口	119
考点 61: 程序查询方式	119
考点 62: 程序中断方式	120
考点 63: DMA 方式	122

第 3 篇 操作系统

第 1 章 操作系统概述	126
真题分布及命题重点	126
考点 64: 操作系统的概念、特征和功能	126
考点 65: 内核态与用户态	127
考点 66: 中断、异常	128
考点 67: 系统调用	128
第 2 章 进程管理	130
真题分布及命题重点	130
考点 68: 进程与线程	130
考点 69: 进程状态与进程控制	131
考点 70: 处理机调度	132
考点 71: 进程同步与互斥	135
考点 72: 经典同步问题	138
考点 73: 死锁	141



第3章 内存管理	144
真题分布及命题重点	144
考点 74: 内存管理的概念	144
考点 75: 连续分配管理方式	145
考点 76: 非连续分配管理方式	146
考点 77: 虚拟页式存储管理	149
考点 78: 抖动	155
第4章 文件管理	157
真题分布及命题重点	157
考点 79: 目录结构	157
考点 80: 文件共享和文件保护	158
考点 81: 文件的操作	159
考点 82: 文件实现	160
考点 83: 磁盘组织与管理	163
第5章 设备管理	165
真题分布及命题重点	165
考点 84: I/O 控制方式	165
考点 85: I/O 软件的层次结构	165
考点 86: I/O 调度与缓冲区	167
考点 87: 设备分配与回收	168

第4篇 计算机网络

第1章 计算机网络体系结构	170
真题分布及命题重点	170
考点 88: 计算机网络的分层结构	170
考点 89: ISO/OSI 参考模型和 TCP/IP 模型	171
第2章 物理层	173
真题分布及命题重点	173
考点 90: 以太网与曼彻斯特编码	173
考点 91: 奈奎斯特定理和香农定理	174
考点 92: 电路交换、报文交换与分组交换	175
考点 93: 物理层接口和物理层设备	176
第3章 数据链路层	177
真题分布及命题重点	177
考点 94: 零比特填充法	177
考点 95: 流量控制、可靠传输与滑动窗口机制	178
考点 96: 随机访问介质访问控制	179
考点 97: 局域网	182
考点 98: 以太网交换机及其工作原理	183

第 1 篇

数据结构

绪论

真题分布及命题重点

年份	分值	单选题	综合题
2009	0+0	-	-
2010	0+√ ^①	-	分析算法的时空复杂度
2011	2+√	分析程度段的时间复杂度	分析算法的时空复杂度
2012	2+√	分析递归程序的时间复杂度	分析算法的时间复杂度
2013	2+√	分析排序程序的时间复杂度	分析算法的时空复杂度

- 命题重点：
- 1. 根据给定代码段的特点，分析算法的时间复杂度。
 - 2. 时空复杂度分析是每年必考考点，或单独命题或结合算法题，务必重点掌握。

考点 1：时间复杂度与空间复杂度

1.（真题 2011）设 n 是描述问题规模的非负整数，下面程序片段的时间复杂度是_____。

```
x=2;
while (x<n/2)
x=2*x;
```

- A. $O(\log_2 n)$ B. $O(n)$ C. $O(n\log_2 n)$ D. $O(n^2)$

【答案】A。

【解析】在程序中，执行频率最高的语句为“ $x=2*x$ ”。设该语句共执行了 $T(n)$ 次，则 $2^{T(n)+1} \leq n/2$ ，故 $T(n) = \log_2(n/2) - 1 = \log_2 n - 2$ ，得 $T(n) = O(\log_2 n)$ 。

【考点笔记】时间复杂度

时间复杂度 $T(n)$ 是指算法中所有语句的频度（执行次数）之和。人们关心的是当 n 趋于无穷时 $T(n)$ 的数量级，而非 $T(n)$ 的准确大小，因此以 $T(n)$ 的数量级来表征时间复杂度。

例如 $T(n) = n^3 + n^2 + n$ ，可认为时间复杂度 $T(n) = O(n^3)$ ，这里数量级最大的一项必定是由最深层循环的语句贡献的，称之为基本运算。由于 $T(n)$ 与算法中基本运算的频度 $f(n)$ 同数量级，所以通常采用基本运算的频度的数量级 $O(f(n))$ 来分析算法的时间复杂度，记为 $T(n) = O(f(n))$ 。

最终归结为一句话：将算法中基本运算的执行次数的数量级作为时间复杂度。

① 注：“+”前为单选题的总分值，每个单选题 2 分；“+”后为综合题的总分值，“√”表示有综合题部分涉及本章知识点。

时间复杂度的计算遵循两种规则:

● **加法法则:** $T(n)=T_1(n)+T_2(n)=O(f(n))+O(g(n))=O(\max(f(n),g(n)))$

● **乘法法则:** $T(n)=T_1(n)\times T_2(n)=O(f(n))\times O(g(n))=O(f(n)\times g(n))$

如, 设 $a\{\}$ 、 $b\{\}$ 、 $c\{\}$ 三个语句块的时间复杂度分别为 $O(1)$ 、 $O(n)$ 、 $O(n^2)$, 则

1) $a\{\$
 $b\{\}$
 $c\{\}$
 $\}$ 的时间复杂度为 $O(n^2)$, 满足加法法则;

2) $a\{\$
 $b\{\$
 $c\{\}$
 $\}$
 $\}$ 的时间复杂度为 $O(n^3)$, 满足乘法法则。

注意: 在有些情况下, 算法中基本运算的执行次数还随问题的输入数据集不同而不同。

2. (真题 2013) 已知两个长度分别为 m 和 n 的升序链表, 若将它们合并为一个长度为 $m+n$ 的降序链表, 则最坏情况下的时间复杂度是_____。

- A. $O(n)$ B. $O(m\times n)$ C. $O(\min(m, n))$ D. $O(\max(m, n))$

【答案】D。

【解析】两个升序链表合并, 两两比较表中元素, 每比较一次确定一个元素的链接位置 (取较小元素, 头插法)。当一个链表比较结束后, 将另一个链表的剩余元素插入即可。最坏的情况是两个链表中的元素依次进行比较, 时间复杂度为 $O(\max(m, n))$ 。

3. (真题 2012) 求整数 $n(n\geq 0)$ 阶乘的算法如下, 其时间复杂度是_____。

```
int fact(int n){
    if(n<=1) return 1;
    return n*fact(n-1);
}
```

- A. $O(\log_2 n)$ B. $O(n)$ C. $O(n\log_2 n)$ D. $O(n^2)$

【答案】B。

【解析】本算法是一个递归运算, 即算法中出现了调用自身的情形。递归的边界条件是 $n\leq 1$, 每调用一次 $fact()$, 传入该层 $fact()$ 的参数值减 1。采用递归式来表示时间复杂度有

$$T(n)=\begin{cases} O(1) & n\leq 1 \\ T(n-1)+1 & n>1 \end{cases}$$

则 $T(n)=T(n-1)+1=T(n-2)+2=\cdots=T(1)+n-1=O(n)$, 故时间复杂度为 $O(n)$ 。

【考点拓展】递归算法

递归是算法领域的一个重要思想。在后续的二叉树中也有递归的应用。递归算法可以通过迭代或栈改写为非递归算法。以本题为例, 也可写成以下两种形式:

迭代方式	栈方式
1) <pre>int fact(int n){ int i,result=1; for(i=n;i>1;i--) result*=i; return result; }</pre>	2) <pre>int fact(int n){ Stack S; int top,result=1; while(n>1) Push(&S,n--); while(Pop(&S,&top)) result*=top; return result; }</pre>

递归算法的特性:

- ① 一个算法直接或间接调用自身。
- ② 必须有一个明确的递归结束条件,称为递归出口。

由于递归算法需要反复进行函数调用与返回,运行效率较低。实现递归的关键是建立递归调用工作栈。当有多个函数构成嵌套调用时,按照后调用先返回的原则,因此递归调用需要开辟栈以存放每一层的返回点、局部变量等,栈的大小也就是递归深度和递归算法空间复杂度的大小。

【考点笔记】空间复杂度

空间复杂度 $S(n)$ 指算法运行过程中所使用的辅助空间的大小,通常结合算法题考查。

若输入数据所占空间只取决于问题本身,和算法无关,则只需分析除输入数据和程序之外的**临时分配的额外空间**。算法**原地工作**是指算法所需辅助空间是常量,即 $O(1)$ 。

线 性 表

真题分布及命题重点

年份	分值	单选题	综合题
2009	0+15	-	单链表的遍历操作
2010	0+13	-	顺序表的循环移动
2011	0+15	-	顺序表的查找/遍历操作
2012	0+13	-	单链表的查找/遍历操作
2013	0+13	-	顺序表的查找/遍历操作

- 命题重点：
- 1. 顺序表的性质、各种操作（查找/遍历/移动/交换/删除等）及灵活应用。
 - 2. 单链表的性质、各种操作（查找/遍历/移动/交换/删除等）及灵活应用。
 - 3. 本章是算法题命题的基础，近 5 年的算法题都是基于线性表的。
 - 4. 考研不会直接考查线性表操作的代码，而往往是带有一定的技巧性。算法题喜欢挑选微软、谷歌等公司的面试题，读者在学习时应注意思维的培养。这也是为什么很多读者熟练记住了线性表的各种操作代码，但遇到此类题时却不知从何下手的缘故。
 - 5. 算法题考查的重点是思想，在写代码时不应过于拘泥于每个细节。对于算法题，即便采用最笨的方法，只要能解决问题且思路和代码正确，也能拿 70% 以上的分数。

考点 2：线性表的顺序表示

1. （真题 2010）设将 n ($n > 1$) 个整数存放到一维数组 R 中。试设计一个在时间和空间两方面都尽可能高效的算法。将 R 中保存的序列循环左移 p ($0 < p < n$) 个位置，即将 R 中的数据由 $(X_0, X_1, \dots, X_{n-1})$ 变换为 $(X_p, X_{p+1}, \dots, X_{n-1}, X_0, X_1, \dots, X_{p-1})$ 。要求：
- 1) 给出算法的基本设计思想。
 - 2) 根据设计思想，采用 C 或 C++ 或 Java 语言描述算法，关键之处给出注释。
 - 3) 说明你所设计算法的时间复杂度和空间复杂度。
- 【解析】考查顺序结构线性表的元素的移动操作，将数组中的序列循环左移。
- 1) 算法的基本设计思想：
- 可以将这个问题看作是 把数组 ab 转换成数组 ba (a 代表数组的前 p 个元素， b 代表数组中余下的 $n-p$ 个元素)，先将 a 逆置得到 $a^{-1}b$ ，再将 b 逆置得到 $a^{-1}b^{-1}$ ，最后将整个 $a^{-1}b^{-1}$ 逆置得到 $(a^{-1}b^{-1})^{-1}=ba$ 。设 Reverse 函数执行将数组元素逆置的操作，对 abcdefgh 向左循环移动 3 ($p=3$) 个位置的过程如下：

Reverse(0,p-1) 得到 cbadefgh;

Reverse(p,n-1) 得到 cbahgfed;

Reverse(0,n-1) 得到 defghabc。

注: Reverse 中, 两个参数分别表示数组中待转换元素的始末位置。

2) 使用 C 语言描述算法如下:

```
void Reverse(int R[],int from,int to) {
    int i,temp;
    for(i=0;i<(to-from+1)/2;i++)
        { temp=R[from+i];R[from+i]=R[to-i];R[to-i]=temp;}
} //Reverse
void Converse(int R[],int n,int p){
    Reverse(R,0,p-1);
    Reverse(R,p,n-1);
    Reverse(R,0,n-1);
}
```

3) 上述算法中 3 个 Reverse 函数的时间复杂度分别为 $O(p/2)$ 、 $O((n-p)/2)$ 和 $O(n/2)$, 故所设计的算法的时间复杂度为 $O(n)$, 空间复杂度为 $O(1)$ 。

【另解】借助辅助数组来实现。

算法思想: 创建大小为 p 的辅助数组 S , 将 R 中前 p 个整数依次暂存在 S 中, 同时将 R 中后 $n-p$ 个整数左移, 然后将 S 中暂存的 p 个数依次放回到 R 中的后续单元。

算法的时间复杂度为 $O(n)$, 空间复杂度为 $O(p)$ 。

【考点笔记】顺序表的定义

顺序表是指用一组连续的存储单元, 依次存储线性表中的各个元素, 从而使得逻辑上相邻的元素在物理位置上也相邻, 因此可以**随机存取**(根据首元素地址和元素序号)表中任一元素。

顺序表的缺点也很明显, 如元素的插入和删除需要移动大量的元素, 插入操作平均需要移动 $n/2$ 个元素, 删除操作平均需要移动 $(n-1)/2$ 个元素, 而且存储分配需要一段连续的存储空间, 不够灵活。

【考点笔记】顺序表的插入操作

对于插入算法, 若表长为 n , 则在第 i 位置插入元素, 则从 a_n 到 a_i 都要向后移动一个位置, 共需移动 $n-i+1$ 个元素, 平均时间复杂度为 $O(n)$ 。代码片段如下:

```
//判断 i 的范围是否有效, 否则非法
//判断当前存储空间是否已满, 否则不能插入
for(int j=L.length;j>=i;j--) //将第 i 个位置及之后的元素后移
    L.data[j]=L.data[j-1];
L.data[i-1]=e; //在位置 i 处放入 e, 数组从 0 开始存储
L.length++; //线性表长度加 1
```

【考点笔记】顺序表的删除操作

对于删除算法, 若表长为 n , 当删除第 i 个元素时, 从 a_{i+1} 到 a_n 都要向前移动一个位置, 共需移动 $n-i$ 个元素, 平均时间复杂度为 $O(n)$ 。代码片段如下:

```
//判断 i 的范围是否有效
for(int j=i;j<L.length;j++) //将第 i 个位置之后的元素前移
    L.data[j-1]=L.data[j];
L.length--; //线性表长度减 1
```

【考点笔记】顺序表的查找

1. 按序号查找, 顺序表具有随机存取(根据首元地址和序号)的特点, 时间复杂度为 $O(1)$ 。
 2. 按值 x 查找, 主要运算是比较操作, 比较的次数与值 x 在表中的位置有关, 也与表长有关, 平均比较次数为 $(n+1)/2$, 时间复杂度为 $O(n)$ 。

2. (真题 2013) 已知一个整数序列 $A = (a_0, a_1, \dots, a_{n-1})$, 其中 $0 \leq a_i < n$ ($0 \leq i < n$)。若存在 $a_{p_1} = a_{p_2} = \dots = a_{p_m} = x$ 且 $m > n/2$ ($0 \leq p_k < n, 1 \leq k \leq m$), 则称 x 为 A 的主元素。例如 $A = (0, 5, 5, 3, 5, 7, 5, 5)$, 则 5 为主元素; 又如 $A = (0, 5, 5, 3, 5, 1, 5, 7)$, 则 A 中没有主元素。假设 A 中的 n 个元素保存在一个一维数组中, 请设计一个尽可能高效的算法, 找出 A 的主元素。若存在主元素, 则输出该元素; 否则输出 -1。要求:

- 1) 给出算法的基本设计思想。
- 2) 根据设计思想, 采用 C 或 C++ 或 Java 语言描述算法, 关键之处给出注释。
- 3) 说明你所设计算法的时间复杂度和空间复杂度。

【解析】1) 给出算法的基本设计思想:

算法的策略是从前向后扫描数组元素, 标记出一个可能成为主元素的元素 Num 。然后重新计数, 确认 Num 是否是主元素。

算法可分为以下两步:

① 选取候选的主元素: 依次扫描所给数组中的每个整数, 将第一个遇到的整数 Num 保存到 c 中, 记录 Num 的出现次数为 1; 若遇到的下一个整数仍等于 Num , 则计数加 1, 否则计数减 1; 当计数减到 0 时, 将遇到的下一个整数保存到 c 中, 计数重新记为 1, 开始新一轮计数, 即从当前位置开始重复上述过程, 直到扫描完全部数组元素。

② 判断 c 中元素是否是真正的主元素: 再次扫描该数组, 统计 c 中元素出现的次数, 若大于 $n/2$, 则为主元素; 否则, 序列中不存在主元素。

2) 算法实现:

```
int Majority(int A[], int n) {
    int i, c, count=1;           //c 用来保存候选主元素, count 用来计数
    c=A[0];                     //设置 A[0] 为候选主元素
    for(i=1; i<n; i++)           //查找候选主元素
        if(A[i]==c)
            count++;             //对 A 中的候选主元素计数
        else
            if (count>0)         //处理不是候选主元素的情况
                count--;
            else{                //更换候选主元素, 重新计数
                c=A[i];
                count=1;
            }
    if(count>0)
        for(i=count=0; i<n; i++) //统计候选主元素的实际出现次数
            if(A[i]==c)
                count++;
    if(count>n/2) return c;      //确认候选主元素
    else return -1;              //不存在主元素
}
```

【1)、2) 的评分说明】①若考生设计的算法满足题目的功能要求且正确, 则 1)、2) 根据所实现