

## 计算机专业（基础综合）模拟试卷 95 (题后含答案及解析)

题型有：1. 单项选择题 2. 综合应用题

单项选择题 1-40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项是最符合题目要求的。

1. 下列叙述中，正确的是( )。I. 非空循环单链表 head 的尾结点 p 满足  $p \rightarrow next = head$  II. 带头结点的循环单链表的头指针为 head，如果  $head \rightarrow next \rightarrow next \rightarrow next = head$  成立，则该单链表的长度为 3 III. 静态链表中的指针表示的是下一个元素在数组中的位置 IV. 将长度为 n 的单链表链接在长度为 m 的单链表之后的算法时间复杂度为  $O(1)$

- A. 仅 I、II、III
- B. I、II、III、IV
- C. 仅 I、III
- D. 仅 I、III、IV

正确答案：C

解析：I：非空循环单链表的尾结点指针应该指向链表头，即  $p \rightarrow next = head$ ，故 I 正确。II：head 指向头结点， $head \rightarrow next$  就指向第一个结点。既然  $head \rightarrow next \rightarrow next \rightarrow next = head$ ，说明此循环链表共有 3 个结点（包含头结点），而单链表中增加头结点仅仅是为了更方便地进行插入和删除操作，它并不存储线性表的元素，故不能算为单链表结点，故此单链表的长度为 2，故 II 错误。III：静态链表中的指针所存储的不再是链表中的指针域，而是其下一个结点在数组中的位置，即数组下标，故 III 正确。IV：将链表连接起来只需  $O(1)$  的操作，但找到具有 m 个结点链表的尾结点需遍历该链表，所以时间复杂度应该为  $O(m)$ ，故 IV 错误。

2. 利用栈求表达式的值时，设立运算数栈 S。假设栈 S 只有两个存储单元，在下列表达式中，不发生溢出的是( )。

- A.  $A - B * (C - D)$
- B.  $(A - B) * C - D$
- C.  $(A - B * C) - D$
- D.  $(A - B) * (C - D)$

正确答案：B

解析：利用栈求表达式的值时，需要设立运算符栈和运算数栈，下面仅举一例。例如，求  $2 \times (5 - 3) + 6 / 2$  的过程如表 6-2 所示。从上述的计算过程中，考生可以自行对 A、B、C、D 选项进行练习，运算数栈 S 的大小分别至少为 4、2、3、3，只有 B 选项满足条件。

3. 设有一个 n 阶三对角线矩阵  $A[n][n]$ ，现把它的三条对角线上的非零元素按行存放到一个一维数组 B 中， $A[1][1]$  存放于 B[1] 中（假定不用 0 下标），那么 B[k] 存放的元素的行号是( )。

- A.  $[(k+1)/3]$
- B.  $[(k+1)/3]$
- C.  $[(k+2)/3]$
- D.  $[(k+2)/3]$

正确答案：B

解析：这种题目最好采用特殊值法，推导过程可能比较繁琐，见表6—3。

4. 已知一棵5阶B—树有53个关键字，并且每个结点的关键字都达到最少状态，则它的深度是( )。

- A. 3
- B. 4
- C. 5
- D. 6

正确答案：B

解析：根据B—树定义，m阶B—树除根结点之外，所有非终端结点至少有 $\lceil m/2 \rceil = 3$ 个子树，即至少有2个关键字。那么在每个结点的关键字最少的情况下，根结点关键字个数为1，其他的结点关键字个数都为2。又第一层有1个结点，第二层有2个结点，第三层有 $2 \times 3$ 个结点，第四层有 $2 \times 3 \times 3$ 个结点。即： $1 \times 1 + 2 \times 2 + 2 \times 3 \times 2 + 2 \times 3 \times 3 \times 2 = 53$ ，根结点加非终端刚好四层，叶子结点那一层不算，故树的深度为4。

5. 下列说法中，正确的是( )。I. 具有10个叶子结点的二叉树中有9个度为2的结点II. 设高度为5的二叉树上只有度为0和度为2的结点，则该二叉树中所包含的结点数至少为9III. 一棵完全二叉树上有1001个结点，则可知叶子结点的个数为501个IV. 高度为h的完全二叉树最少有 $2^h$ 个结点

- A. 仅I、II
- B. 仅II、III、IV
- C. 仅I、II、IV
- D. 仅I、II、III

正确答案：D

解析：I：二叉树叶子结点的个数比度为2的结点的个数多1，故I正确。  
 总结：这个性质在选择题中常有体现（见下面的补充例题），并且需要灵活运用。比如题目可能问，二叉树中总的结点数为n，则树中空指针的个数是多少？我们可以将所有的空指针看作叶子结点，则图中原有的所有结点都成了双分支结点。因此可得空指针域的个数为树中所有结点个数加1，即 $n+1$ 个。这个性质还可以扩展，即在一棵度为m的树中，度为1的结点数为 $n_1$ ，度为2的结点数为 $n_2$ ……度为m的结点数为 $n_m$ ，则叶子结点数 $n_0 = 1 + n_2 + 2n_3 + \dots + (m-1)n_m$ 。推导过程如下：总结点 $= n_0 + n_1 + n_2 + n_3 + \dots + n_m$ ……①总分支数 $= 1 \times n_1 + 2 \times n_2 + \dots + m \times n_m$ （度为m的结点引出m条分支）……②总分支数=总结点数-1……③将式①和式②代入式③并化简得 $n_0 = 1 + n_2 + 2n_3 + \dots + (m-1)n_m$   
 补充例题：在一棵二叉树中度为0的结点个数为k，度为1的结点个数为m，则

该二叉树采用二叉链存储结构时,有( )个指针指向孩子结点。A.  $k$

B.  $m$

C.  $2k+m-2$

D.  $2k+m$  C. 本题考查树的链式存储结构。首先,由二叉树的性质可知, $n_0=n_2+1$  (多次用到,考生一定要记住!),得到  $n_2=k-1$ 。其次,二叉树的结点总数  $n=n_0+n_1+n_2=2k+m-1$ 。求指向孩子结点的指针个数其实就是求该二叉树的分支数,而分支数就是等于总结数-1,所以答案为  $2k+m-2$ ,故选 C 选项。II: 最少结点的情况应该是除根结点层只有 1 个结点外,其余 4 层都有 2 个结点,因此结点总数为  $2 \times (5-1) + 1 = 9$ 。如图 6-4 所示,故 II 正确。总结: 设高度为  $h$  的二叉树只有度为 0 和度为 2 的结点,则此类二叉树中所包含的结点数至少为  $2h-1$ 。m: 由二叉树的性质可知:  $n_0=n_2+1$ , 且完全二叉树度为 1 的结点个数要么为 0, 要么为 1。又因为二叉树的总结点个数  $n=n_0+n_1+n_2$ 。将  $n_0=n_2+1$  代入, 可得  $n=2n_0+n_1-1$ ; 由于  $n=1001$ , 得到  $2n_0=1002+n_1$ 。①当  $n_1=1$  时, 无解。②当  $n_1=0$  时, 可解得  $n_0=501$  故 III 正确。IV: 高度为  $h$  的完全二叉树中, 第 1 层~第  $h-1$  层构成一个高度为  $h-1$  的满二叉树, 结点个数为  $2^h-1-1$ 。第  $h$  层至少有一个结点, 所以最少的结点个数  $=(2^h-1-1)+1=2^h-1$ , 故 IV 错误。

6. 在平衡二叉树中插入一个结点就造成了不平衡, 设最低的不平衡结点为 A, 并已知 A 的左孩子的平衡因子为 -1, 右孩子的平衡因子为 0, 则为使其平衡, 应做( )型调整。

A. LL

B. RR

C. RL

D. LR

正确答案: D

解析: 既然最低不平衡结点是 A, 则以 A 为根的子树不平衡的情况有 4 种, 如图 6-5 所示。又因为 A 的左孩子的平衡因子为 -1, 右孩子的平衡因子是 0, 只有第 2 个符合, 所以应当做 LR 型调整。【总结】为了不至于混淆调整不平衡状态时做出的是什么类型的调整, 以下介绍一种简便的方法: 找出最低的不平衡结点到刚刚插入之后 (导致不平衡) 的结点的路径, 这种路径的序列也就标识了应该做出什么类型的调整, 如图 6-5 的 2 所示, 最低不平衡结点到插入结点的路径序列是 LR, 那么就应该做 LR 调整。

7. 下列关于无向图的说法中, 正确的是( )。I. 无向图中某个顶点的度是指图中与该顶点连通的顶点数 II. 在一个具有  $n$  个顶点的无向图中, 要连通全部顶点至少需要  $n-1$  条边 III. 无向图的邻接矩阵是对称矩阵 IV. 具有  $n$  个顶点的无向图, 最多有  $n$  个连通分量

A. 仅 I、II、III

B. 仅 II、II I、IV

C. 仅 III

D. I、II、III、IV

正确答案: B

解析：I：无向图顶点的度即为一个顶点所引出边的条数，等价于一个顶点所含有的邻接顶点的个数，而不是与该顶点连通的顶点数（这样就会扩大范围，如图6—6所示），故I错误。顶点V2的度应该是1，而如果度是按照图6—6中与该顶点连通的顶点数来定义，顶点V2的度应该是3，明显错误。II： $n$ 个顶点的无向图要连通的话只需每个顶点做一个结点，构成一棵树即可（解题关键），并且此时是边最少的情况。对于树来说，顶点的个数比边要多1，故II正确。III：显然，在无向图中，每条边（没有方向）对应于矩阵中与主对角线对称的两个“1”，因此无向图对应的邻接矩阵是对称的，故III正确。IV：无向图的连通分量最少只有一个，即其自身；最多有 $n$ 个，即该图没有边，则每个顶点构成一个连通分量，故IV正确。

8. 下列关于强连通图的说法中，正确的是( )。I.  $n$ 个顶点构成的强连通图至少有 $n$ 条边 II. 强连通图是任何顶点到其他所有顶点都有边 III. 完全有向图一定是强连通图

- A. 仅 I、II
- B. 仅 II、III
- C. 仅 I、III
- D. I、II、III

正确答案：C

解析：I：强连通图是相对于有向图而言的，即在有向图G中，任何两个顶点都存在路径。所以最少的情况应该是 $n$ 个顶点构成一个首尾相连的环，共有 $n$ 条边，故I正确。II：这个选项不细心的话很容易误选。在有向图中，边和路径是不同的概念。有向图中顶点A和B之间存在边，不能说明A和B是互相连通的，所以说正确的表述应该是强连通图是任何顶点到其他所有顶点都有路径，故II错误。III：完全有向图肯定是任何顶点到其他所有顶点都有路径，故III正确。

9. 假设初始为空的散列表的地址空间为(0..10)，散列函数为 $H(\text{key}) = \text{key} \bmod 11$ ，采用线性探测再散列法处理冲突，若依次插入关键字37、95、27、14、48，则最后一个关键字值48的插入位置是( )。

- A. 4
- B. 5
- C. 6
- D. 8

正确答案：C

解析：首先通过散列函数 $H(\text{key}) = \text{key} \bmod 11$ 的计算得知，37、95、27、14分别插入到散列表中的4、7、5、3的位置。而 $48 \bmod 11 = 4$ ，但是此时4已经有元素了，根据线性探测再散列法处理冲突的原则，依次探测位置4的下一个地址，直到此地址为空，发现6为空则插入，故选C选项。补充：如果此题改为使用平方探测法，则又应该选择哪一个选项？解析：平方探测法的原理是设发生冲突的地址为 $d$ ，则平方探测法的探测序列为 $d+1^2, d-1^2, d+2^2, d-2^2, \dots$ 。位置4不空时，下一个探测的位置应该为5，发现又不空，则下一个探测的位置应该是3，发现又不空。接着再探测位置8，发现为空，将元素插入，故选D选项。平

方探测法是一种较好的处理冲突的方法，可以避免出现堆积问题。它的缺点是不能探测到散列表上的所有单元，但至少能探测到一半单元。

10. 设待排序元素序列所有元素的排序码都相等，则下列排序方法中排序速度最慢的是( )。

- A. 直接插入排序
- B. 起泡排序
- C. 简单选择排序
- D. 基数排序

正确答案：C

解析：当所有待排序元素的排序码都相等时，直接插入排序的排序码比较次数为  $n-1$ ，元素移动次数为 0；起泡排序的排序码比较次数为  $n-1$ ，元素移动个数为 0；简单选择排序的排序码比较次数为  $n(n-1)/2$ ，元素移动次数为 0；基数排序采用静态链表存储待排序元素，用于分配的桶亦采用链式队列，排序码比较次数为  $n \times d$  ( $d$  是排序码位数)，元素移动次数为 0，故排序速度最慢的是简单选择排序。

11. 假设有 5 个初始归并段，每个归并段有 20 个记录，采用 5 路平衡归并排序，若采用败者树的方法，总的排序码比较次数不超过 ( )。

- A. 20
- B. 300
- C. 396
- D. 500

正确答案：B

解析：假设采用  $k$  路平衡归并排序算法，则败者树的高度为  $\lceil \log_2 k \rceil + 1$ 。在每次调整后，找下一个具有最小排序码记录时，最多做  $\lceil \log_2 k \rceil$  次排序码比较。由题意可知，总共有 100 个记录，所以总的比较次数不超过  $100 \times \lceil \log_2 5 \rceil = 300$ 。注意：采用败者树进行  $k$  路平衡归并的外部排序算法，其总的归并效率与  $k$  无关。

12. 下列说法中，错误的是( )。I. 设浮点数的基数为 4，尾数用原码表示，则 0.000 010 为规格化数 II. 浮点数运算中，运算结果超出尾数表示范围则表示溢出 III. 任何情况下，浮点数的右规操作最多只会进行一次

- A. 仅 I、III
- B. 仅 II、III
- C. 仅 I、II
- D. I、II 和 III

正确答案：C

解析：I：对于原码表示的基值为 4 的小数，规格化的形式是小数点后 2 位不全为 0，故 I 错误。最笨的解题思路：基数  $r=4$ ，由于  $1/r \leq |M| \leq 1$ ，即尾数的十进制绝对值在 0.25~1 之间。而  $(0.000 010)_2 = 0.03125$  不是规格化数。II：浮点数的溢出并不是由尾数来判断的，而是规格化后阶码超出所能表示的范围

时,才表示溢出,故 II 错误。III:在浮点数的运算过程中,尾数如果出现  $01.XXX\cdots X$  和  $10.XXX\cdots X$ ,则需要进行右规,并且只需进行一次右规尾数就会变成规格化数,但是左规操作可能不止一次,故 III 正确。

13. 下列关于定点数原码一位乘法的描述中,错误的是( )。I. 符号位不参加运算,根据数值位的乘法运算结果确定结果的符号位 II. 在原码一位乘算法过程中,所有的移位均是算术移位操作 III. 假设两个  $n$  位数进行原码一位乘,部分积至少需要使用  $n$  位寄存器

- A. 仅 I、II
- B. 仅 II、III
- C. 仅 I、III
- D. I、II、III

正确答案: D

解析: I: 在原码一位乘算法过程中,符号位是不参加运算的,结果的符号位是被乘数的符号位和乘数的符号位异或的结果,故 I 错误。II: 在原码一位乘算法过程中,由于参与操作的数是真值的绝对值,所以没有正负可言,故在原码一位乘法中运算过程中所有的移位均是逻辑移位操作,即在高位添加 0,故 II 错误。III: 由于在部分积相加中,可能导致两个小数相加大于 1,所以部分积至少需要使用  $n+1$  位寄存器,故 III 错误。

14. 某容量为 256MB 的存储器由若干 16Mx8bit DRAM 芯片构成,该 DRAM 芯片的地址引脚和数据引脚总数是( )。

- A. 20
- B. 24
- C. 32
- D. 36

正确答案: A

解析: 很多不了解 DRAM 引脚结构的同学很可能会得出  $24+8=32$  的结果,其实这是不正确的,在《高分笔记》当中讲过半导体存储芯片的译码驱动方式,其中介绍了重合法,将存储单元分成行和列,然后分别通过行地址线和列地址线来确定行列地址从而确定一个单元,这里 DRAM 采用引脚复用,将行地址线和列地址线合用作一组,只不过在译码时,需要发送两次地址信号(相当于一行地址,一次列地址),从而减少了 DRAM 的引脚总数,便于设计 DRAM; 因此这里地址空间是 16M,需要 24 个地址位来标识,分为两次发送,则地址引脚数为 12,故地址引脚和数据引脚总数为  $12+8=20$ 。【总结】DRAM 芯片采用引脚复用,且行列地址位数一致。

15. 现有一  $64K \times 2\text{bit}$  的存储器芯片,欲设计具有同样存储容量的存储器,有( )种方法可以合理地安排地址线 and 数据线引脚的数目,且使两者之和最小。

- A. 2
- B. 3

- C. 4
- D. 5

正确答案：A

解析：不妨设地址线和数据线的数目分别为  $x$  和  $y$ 。只需要满足  $2x \times y = 64K \times 2$ ，所以就有如下方案：当  $y=1$  时， $x=17$ ；当  $y=2$  时， $x=16$ ；当  $y=4$  时， $x=15$ ；当  $y=8$  时， $x=14$ ；后面的就不要计算了，肯定比前面的引脚数目多。从以上分析可以看出，当数据线分别为 1 或 2 时，地址线和数据线引脚的数目之和为 18，达到最小，并且有两种解答。

16. 某计算机有 30 个通用寄存器，采用 32 位定长指令字，操作码字段（不含寻址方式）为 8 位，Add 指令的源操作数和目的操作数分别采用寄存器直接寻址和基址寻址方式。若基址寄存器可使用任一通用寄存器，且偏移量用补码表示，则 Add 指令中偏移量的取值范围是（ ）。

- A.  $-4096 \sim 4095$
- B.  $-2048 \sim 2047$
- C.  $-1023 \sim 1024$
- D.  $-3071 \sim 3072$

正确答案：B

解析：首先可以直接排出 C、D 选项，因为无论偏移量是多少位，由于偏移量是采用补码表示的，根据补码的特性，它比源码表示的数多一位，而且多出来的就是补码的最小值。因此偏移量的最小值一定是一个偶数。操作码占 8 位，两个操作数具有两种不同的寻址方式，则需要 2 位寻址特征位，另外一共有 30 个寄存器，故需要 5 位来标识选择哪个寄存器，所以偏移量的位数  $= 32 - 8 - 2 - 5 - 5 = 12$ ，而 12 位的带符号的补码所能表示的数的范围为  $-2048 \sim 2047$ 。【提示】在考场上有时候即使我们不能一步就算出结果，或者题目复杂的时候，可以抓住问题的一些细节来排除某些选项，这对我们分析余下的选项也是很有帮助的。

17. 与本指令的地址有关的寻址方式是（ ）。
- A. 寄存器寻址
  - B. 直接寻址
  - C. 相对寻址
  - D. 间接寻址

正确答案：C

解析：相对寻址本身就是相对于本指令地址进行上下浮动，所以相对寻址的区间范围和本指令的地址密切相关，其他 3 个选项都与本指令的地址无关。

18. 假定执行最复杂的指令需要完成 6 个子功能，分别由对应的功能部件 A~F 来完成，每个功能部件所花的时间分别为 80ns、40ns、50ns、70ns、20ns、30ns，流水线寄存器延时为 20ns，现把最后两个功能部件 E 和 F 合并，以产生一个五段流水线。该五段流水线的时钟周期至少是（ ）。

- A. 70ns

- B. 80ns
- C. 90ns
- D. 100ns

正确答案：D

解析：指令的各个子功能在不同的部件中是并行执行的，因此执行这条指令的时间一定是各个子功能中所花的最长时间，当前最长时间为 80ns，当合并 E 和 F 这两个功能部件之后，合并子功能执行时间为 50ns，因此最长的时间还是 80ns，再加上 20ns 的寄存器延迟，所以五段流水线的时钟周期至少是 100ns。

19. 在微程序控制器中，执行指令微程序的首条微指令地址是由( )得到的。

- A. 程序计数器 PC
- B. 前条微指令
- C.  $\mu\text{PC}+1$
- D. 指令操作码映射

正确答案：D

解析：本题问的是微程序中首条微指令的地址，稍不注意就可能误选 B，微程序是用来解释指令的，通过指令操作码的内容来区别指令，然后根据指令操作码映射找到对应解释这个指令的微程序段。因此首条微指令的地址是由指令操作码映射而来的。

20. 指令流水线中出现数据相关时流水线将受阻，( )可解决数据相关问题。

- A. 增加硬件资源
- B. 采用旁路电路技术
- C. 采用分支预测技术
- D. A~C 都可以

正确答案：B

解析：在流水线处理器中处理数据相关问题有两种方法：一种是暂停相关指令的执行，即暂停流水线，直到能够正确读出寄存器操作数为止；另一种是采用旁路电路技术，即采用专门的数据通路，直接把结果送到 ALU 的输入端，也就是把内部数据前推，即不必等待某条指令的执行结果写回到寄存器后，再从寄存器取出结果，而是直接将执行结果通过专用通路送至需要该结果的地方。

21. 在计数器定时查询方式下，若每次计数从  $[n/2]$  开始，则( )。

- A. 设备号小的优先级高
- B. 每个设备使用总线的机会相等
- C. 设备号大的优先级高
- D. 以上说法都不正确

正确答案：D

解析：当每次计数从 $\lfloor n/2 \rfloor$ 开始时，所有设备被分为两部分，设备号为 $\lfloor n/2 \rfloor$ 到  $n$  的设备优先级高于设备号为 0 到 $\lfloor n/2 \rfloor - 1$  的设备；且在这两部分内，却是设备小的优先级高，故 A、B、C 选项都是错误的。

22. 以下 4 个步骤在通道过程中的正确顺序是( )。I .组织 I/O 操作 II . 向 CPU 发出中断请求 III . 编制通道程序 IV . 启动 I/O 通道

- A. I → II → III → IV
- B. II → III → I → IV
- C. IV → III → II → I
- D. III → IV → I → II

正确答案：D

解析：通道的工作过程如下：(1)用户程序中使用访管指令进入操作系统的管理程序，由 CPU 通过管理程序组织一个通道程序，并使用 I/O 指令启动通道（此后 CPU 就可以并行运行应用程序了）。(2)通道并行执行 CPU 为它组织的通道程序（通道程序在主存中），完成指定的数据输入输出工作。(3)通道程序结束后向 CPU 发出中断请求。CPU 响应这个中断请求后，第二次调用管理程序对输入输出中断请求进行处理。这样，每完成一次输入输出工作，CPU 只需要两次调用管理程序，大大减少了对用户程序的打扰。补充：在采用通道结构的系统中，也需要使用 I/O 指令，但这种 I/O 指令比较简单，它并不直接控制具体 I/O 操作，只是负责通道的启动和停止、查询通道或设备的状态，从而控制通道去完成 I/O 操作。

23. 下列关于批处理技术和多道程序设计技术说法中，正确的是( )。  
I . 批处理系统的最主要缺点是不能并发执行 II . 所谓多道程序设计，是指每一个时刻有若干个进程在执行 III . 引入多道程序设计的前提条件之一是系统具有中断功能 IV , 采用多道程序设计的系统中，系统的程序道数越多，系统的效率越高

- A. 仅 I、II
- B. 仅 II、III
- C. 仅 III
- D. 仅 I、IV

正确答案：C

解析：I 错误，批处理系统的最主要缺点是缺乏交互性。I 的表述肯定是错的，多道批处理系统就可以并发执行多个程序。这里多道是指允许多个进程同时驻留在主存中，按照某种原则分派处理机，逐个执行这些程序。这里其实还考查了并发的概念。并行性是指两个或多个事件在同一时刻发生；而并发性是指两个或多个事件在同一时间间隔内发生。II 错误，多道程序设计是指把多个程序同时存放在内存中，使它们同时处于运行状态。但是，在单处理机环境中，同一时刻只有一个进程在执行。知识点回顾：多道程序设计技术的主要特点是多道、宏观上并行、微观上串行。多道是指计算机内存中同时存放多个相互独立的程序。宏观上并行是指同时进入系统中的多道程序都处于运行过程中（即同时存放在内存中）。微观上串行是指在单处理机环境中，内存中的多道程序轮流占有 CPU，交替执行。III 正确，有了中断后才能实现进程间并发，进程间并发才有可能把多个

IV 错误，程序道数如果过多的话，会导致每个程序分配到的内存不够，很多程序所需的程序和代码需要临时从磁盘调入到内存，系统会频繁地处于 I/O 状态中，导致系统效率降低。

24. 假设系统中所有进程是同时到达，则最不利于短作业的进程调度算法是( )。

- A. FCFS
- B. SPF
- C. RR
- D. 高响应比优先

正确答案：A

解析：本题可用排除法。首先排除 B 选项。因为它是短作业优先算法，肯定是有利于短作业的。然后继续排除 C 选项。RR 兼顾长短作业，一般来说在时间片不是的太长的情况下，对于短作业还是比较公平的。（时间片设的无限长，即变成了 FCFS 算法。）最后排除 D 选项。响应比=作业响应时间/作业执行时间=（作业执行时间+作业等待时间）/作业执行时间=1+作业等待时间/作业执行时间。在作业等待时间相同的情况下，短作业的响应比是更高的，所以高响应比优先有利于短作业。综上分析，本题选 A 选项。知识点回顾：表 6—4 给出几种常见的进程调度算法特点的总结，读者要在理解的基础上识记。

25.  $P_i(\quad)\{\text{Lock}(m\_mutex); \quad / \text{含义为获取互斥信号量 } a=\text{new int}[100];$   
 $/ \text{开辟一个大小为 } 100 \text{ 的整型数组空间, } / \text{并用全局指针变量 } a \text{ 保存空间地址}$   
 $\text{UnLock}(m\_mutex); \text{ free}(a); \quad / \text{释放数组空间, 且 } a \text{ 的值不改变}\}$  有多个优先级相同的进程  $P_i$ 。试问下列同时运行多个进程  $P_i$ ，可能会出现错误的是( )。

- A. 内存泄露
- B. 内存越界访问
- C. 内存泄露和内存越界访问
- D. 无

正确答案：C

解析：由于  $a$  为全局指针变量，即属于临界资源，访问  $a$  的代码都属于临界区，临界区应该在  $\text{Lock}(m\_mutex)$  和  $\text{UnLock}(m\_mutex)$  之间，使各个进程互斥访问  $a$ 。但由于本题  $\text{free}(a)$  在  $\text{Lock}(m\_mutex)$  和  $\text{UnLock}(m\_mutex)$  之外，所以是会出现错误的。举例：假设有进程  $P_1$  和  $P_2$ ， $P_1$  进程申请的数组空间地址赋给  $a$  之后，还没有  $\text{free}$  掉。 $P_2$  进程又申请了新的数组空间又把地址赋给  $a$ ，导致  $P_1$  进程申请的空间地址丢失（即内存泄露）。然后  $P_1$  进程继续执行， $P_1$  进程执行  $\text{free}$  操作，将  $P_2$  进程申请的空间释放掉了， $P_2$  进程继续执行， $P_2$  进程执行  $\text{free}$  操作， $\text{free}$  操作访问了不属于  $P_2$  进程的空间（之前已经被  $P_1$  释放掉了），会发生内存越界访问。知识点扩展：内存泄露：当以前分配的一片内存不再需要使用或无法访问时，但是并没有释放它，那么对于该进程来说，会因此导致总可用内存的减少，这时就出现了内存泄漏。内存越界访问：简单地说，进程访问了不属于该进程的内存空间。

26 生产者进程和消费者进程代码如下。生产者进程有一个局部变量 `nextProduced` 以存储新产生的新项：`while (1){/*produce an item in nextProduced*/while((in+1) % BUFFER_SIZE==out); /*do nothing*/buffer[in]=nextProduced; in=(in+1) %BUFFER_SIZE; .}`消费者进程有一个局部变量 `nextConsumed`，以存储所要使用的项：`while (1){while (in==out); /*do nothing*/nextConsumed=buffer[ out]; out=(out+1) %BUFFER_SIZE; /*consume the item in nextConsumed*/}`当 `in==out` 和 `(in+1)%BUFFER_SIZE==out` 条件成立的时候，缓冲区中 `item` 数目各是( )。

- A. 0, BUFFER\_SIZE
- B. 0, BUFFER\_SIZE—1
- C. BUFFER\_SIZE—1, 0
- D. BUFFER\_SIZE, 0

正确答案：B

解析：通过阅读代码可知，变量 `in` 指向缓冲区中下一个空位，变量 `out` 指向缓冲区中的第一个非空位。`BUFFER_SIZE` 是缓冲区最大能容纳的 `item` 数目。`buffer` 中，非空的位置范围是 `[out, in—1]` 或者 `[out, BUFFER_SIZE—1] U [0, in—1]`，即有如图 6—7 所示的两种情况。当 `in==out` 时，前一个操作肯定是运行了消费者进程（`out` 追上了 `in`），因为生产者进程中，当遇到 `(in+1) % BUFFER_SIZE==out` 时就忙等，即生产进程无法使 `in==out`，所以此时缓冲区中 `item` 数目应该是 0。当 `(in+1)%BUFFER_SIZE==out` 时，即 `in` 差一个空位就追上 `out` 了，此时缓冲区中 `item` 数目应该是 `BUFFER_SIZE—1`。所以本题正确答案是 B 选项。

27. 某操作系统采用可变分区分配存储管理方法，操作系统占用低地址部分的 126KB。用户区大小为 386KB，且用户区始址为 126KB，用空闲分区表管理空闲分区。若分配时采用分配空闲区高地址的方案，且初始时用户区的 386KB 空间空闲，对下述申请序列：作业 1 申请 80KB，作业 2 申请 56KB，作业 3 申请 120KB，作业 1 完成并释放空间，作业 3 完成并释放空间，作业 4 申请 156KB，作业 5 申请 80KB。如果用首次适应算法处理上述序列，最后的空闲分区的首地址为( )。

- A. 126
- B. 432
- C. 256
- D. 220

正确答案：A

解析：本题需要注意的有，一般首次适应算法是要求空闲分区链以地址递增的次序链接，本题相反，是以地址递减的顺序链接的。为描述方便，本题用“(分区首址，分区长度)”的形式描述系统中的分区。由题中所给条件可知，最初系统中只有一个空闲区，大小为 386KB，始址为 126KB，即(126KB, 386KB)。采用首次适应算法的操作流程如表 6—5 所示。

28. 在分页式系统中，分页由( )实现。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/858101020065006120>