

## 公共基础第三十套试卷

1、下列关于栈的描述中错误的是\_\_\_\_\_。

- A. 栈是先进后出的线性表
- B. 栈只能顺序存储
- C. 栈具有记忆作用
- D. 对栈的插入与删除操作中，不需要改变栈底指针

【答案】B

【解析】本题考查的是栈和队列。栈是一种特殊的线性表，这种线性表只能在固定的一端进行插入和删除操作，允许插入和删除的一端称为栈顶，另一端称为栈底。一个新元素只能从栈顶一端进入，删除时，只能删除栈顶的元素，即刚刚被插入的元素。所以栈又称先进后出表（FILO-First In Last Out）。线性表可以顺序存储，也可以链式存储，而栈是一种线性表，也可以采用链式存储结构。

2、对于长度为n的线性表，在最坏情况下，下列各排序法所对应的比较次数中正确的是\_\_\_\_\_。

- A. 冒泡排序为  $n/2$
- B. 冒泡排序为  $n$
- C. 快速排序为  $n$
- D. 快速排序为  $n(n-1)/2$

【答案】D

【解析】本题考查的是基本排序算法。

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  $n/2$  遍的从前往后扫描和  $n/2$  遍的从后往前扫描，需要比较次数为  $n(n-1)/2$ 。快速排序法的最坏情况比较次数也是  $n(n-1)/2$ 。

3、对长度为  $n$  的线性表进行顺序查找，在最坏情况下所需要的比较次数为\_\_\_\_\_。

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

【答案】C

【解析】本题考查的是顺序查找。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，或者被查找的元素根本就不在线性表中，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。所以对长度为  $n$  的线性表进行顺序查找，在最坏情况下需要比较  $n$  次。

4、下列关于栈的描述正确的是\_\_\_\_\_。

- A. 在栈中只能插入元素而不能删除元素
- B. 在栈中只能删除元素而不能插入元素
- C. 栈是特殊的线性表，只能在一端插入或删除元素
- D. 栈是特殊的线性表，只能在一端插入元素，而在另一端删除元素

【答案】C

【解析】栈实际上也是线性表，只不过是一种特殊的线性表。在这种特殊的线性表中，其插入和删除只在线性表的一端进行。

5、下列叙述中正确的是\_\_\_\_\_。

- A. 一个逻辑数据结构只能有一种存储结构
- B. 数据的逻辑结构属于线性结构，存储结构属于非线性结构

- C. 一个逻辑数据结构可以有多种存储结构，且各种存储结构不影响数据处理的效率
- D. 一个逻辑数据结构可以有多种存储结构，且各种存储结构影响数据处理的效率

【答案】D

【解析】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，常用的存储结构有顺序、链接、索引等存储结构。而采用不同的存储结构，其数据处理的效率是不同的。

6、按照"后进先出"原则组织数据的数据结构是\_\_\_\_\_。

- A. 队列
- B. 栈
- C. 双向链表
- D. 二叉树

【答案】B

【解析】栈和队列都是一种特殊的操作受限的线性表，只允许在端点处进行插入和删除。二者的区别是：栈只允许在表的一端进行插入或删除操作，是一种"后进先出"的线性表；而队列只允许在表的一端进行插入操作，在另一端进行删除操作，是一种"先进先出"的线性表。

7、下列描述中正确的是\_\_\_\_\_。

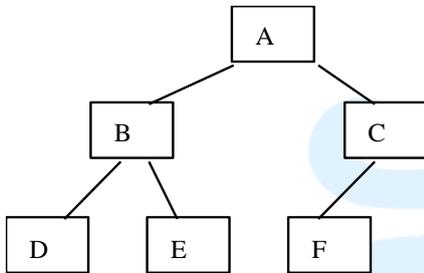
- A. 线性链表是线性表的链式存储结构
- B. 栈与队列是非线性结构
- C. 双向链表是非线性结构
- D. 只有根结点的二叉树是线性结构

【答案】A

【解析】根据数据结构中各数据元素之间前后关系的复杂程度，一般将数据结构分为两大类型：线性结构与非线性结构。如果一个非空的数据结构满足下列两个条件：①有且只有一个

根结点；②每个结点最多有一个前件，也最多有一个后件。则称该数据结构为线性结构，又称线性表。所以线性表、栈与队列、线性链表都是线性结构，而二叉树是非线性结构。

8、对如下二叉树



进行后序遍历的结果为\_\_\_\_\_。

- A. ABCDEF
- B. DBE AFC
- C. ABDECF
- D. DEBFCA

【答案】D

【解析】后序遍历指在访问根结点、遍历左子树与遍历右子树这三者中，首先遍历左子树，然后遍历右子树，最后访问根结点；并且遍历左、右子树时，仍然先遍历左子树，然后遍历右子树，最后访问根结点。

9、下列叙述中正确的是\_\_\_\_\_。

- 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

【答案】D

【解析】算法的复杂度主要包括时间复杂度和空间复杂度：算法在运行过程中需辅助存储空间的大小称为算法的空间复杂度；算法的时间复杂度是指执行算法所需要的计算工作量，即算法执行过程中所需要的基本运算次数，为了能够比较客观地反映出算法的效率，在度量一个算法的工作量时，不仅应该与所使用的计算机、程序设计语言以及程序编制者无关，而且还应该与算法实现过程中的许多细节无关。为此，可以用算法在执行过程中所需基本运算的执行次数来度量算法的工作量。

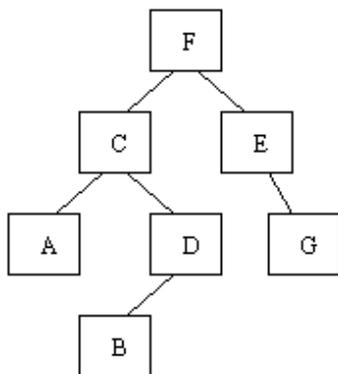
10、在长度为 64 的有序线性表中进行顺序查找，最坏情况下需要比较的次数为\_\_\_\_\_。

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

【答案】B

【解析】在进行顺序查找过程中，如果线性表中的第1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，或者被查找的元素根本就不在线性表中，则为了查找这个元素需要与线性表中所有的元素进行较，这是顺序查找的最坏情况。所以对长度为n的线性表进行顺序查找，在最坏情况下需要比较n次。

11、对下列二叉树



进行中序遍历的结果是\_\_\_\_\_。

- A. ACBDFEG
- B. ACBDFGE
- C. ABDCGEF
- D. FCADBEG

【答案】A

【解析】二叉树的中序遍历指在访问根结点、遍历左子树与遍历右子树这三者中，首先遍历左子树，然后访问根结点，最后遍历右子树；并且遍历左、右子树时，仍然先遍历左子树，然后访问根结点，最后遍历右子树。

12、 下列叙述中正确的是\_\_\_\_\_。

- A. 数据的逻辑结构与存储结构是一一对应的
- 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. 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
- D. 算法的时间复杂度与空间复杂度一定相关

【答案】B

【解析】我们通常用时间复杂度和空间复杂度来衡量算法效率，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，而一个算法的空

间复杂度，一般是指执行这个算法所需要的内存空间；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。

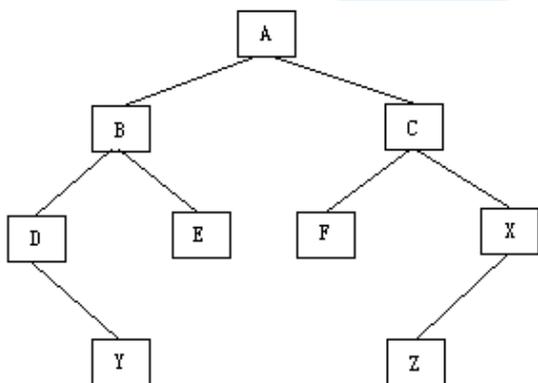
13、下列对队列的叙述正确的是\_\_\_\_\_。

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

【答案】D

【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，允许插入的一端称为队尾，允许删除的一端称为队头。它又称为"先进先出"或"后进后出"的线性表，体现了"先来先服务"的原则。

14、对下列二叉树



进行前序遍历的结果为\_\_\_\_\_。

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

【答案】C

【解析】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3种：前序遍历、中序遍历和后序遍历。前序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中，首先访问根结点，然后遍历左子树，最后遍历右子树；并且遍历左、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。

15、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\_\_\_\_\_。

A. $n+1$

B. $n-1$

C. $2n$

D. $n/2$

【答案】A

【解析】二叉树的性质3：在任意一棵二叉树中，度为0的结点（即叶子结点）总是比度为2的结点多一个。本题中度为2的结点数为 $n$ ，故叶子结点数为 $n+1$ 个。

16、冒泡排序在最坏情况下的比较次数是\_\_\_\_\_。

A. $n(n+1)/2$

B.  $n\log_2 n$

C. $n(n-1)/2$

D. $n/2$

【答案】C

【解析】冒泡排序法是一种最简单的交换类排序方法，它是通过相邻数据元素的交换逐步将线性表变成有序。

假设线性表的长度为 $n$ ，则在最坏的情况下，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 $n(n-1)/2$ 。

17、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\_\_\_\_\_。

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

【答案】A

【解析】二叉树的性质3：在任意一棵二叉树中，度为0的结点（即叶子结点）总是比度为2的结点多一个。本题中度为2的结点数为 $70-1=69$ ，该二叉树中的总结点数为 $70+80+69=219$ 。

18、算法的有穷性是指\_\_\_\_\_。

- A. 算法程序的运行时间是有限的
- B. 算法程序所处理的数据量是有限的
- C. 算法程序的长度是有限的
- D. 算法只能被有限的用户使用

【答案】A

【解析】算法的有穷性，是指算法必须在有限的时间内做完，即算法必须能在执行有限个步骤之后终止。

19、对长度为n的线性表排序，在最坏情况下，比较次数不是 $n(n-1)/2$ 的排序方法是\_\_\_\_\_。

- A. 快速排序
- B. 冒泡排序

C. 直接插入排序

D. 堆排序

【答案】D

【解析】对于长度为 $n$ 的线性表，在最坏的情况下，快速排序所需要的比较次数为 $n(n-1)/2$ ；冒泡排序所需要的比较次数为 $n(n-1)/2$ ；直接插入排序所需要的比较次数为 $n(n-1)/2$ ；堆排序所需要的比较次数为 $O(n\log_2 n)$ 。

20、一个栈的初始状态为空。现将元素1、2、3、4、5、A、B、C、D、E依次入栈，然后再依次出栈，则元素出栈的顺序是\_\_\_\_\_。

A. 12345ABCDE

B. EDCBA54321

C. ABCDE12345

D. 54321EDCBA

【答案】B

【解析】栈是一种特殊的线性表，这种线性表只能在固定的一端进行插入和删除操作，允许插入和删除的一端称为栈顶，另一端称为栈底。一个新元素只能从栈顶一端进入，删除时，只能删除栈顶的元素，即刚刚被插入的元素。这表明栈的运算规则是“先进后出”（或称“后进先出”）。在栈顶进行插入运算，称为进栈（或入栈），在栈顶进行删除运算，称为退栈（或出栈）。本题中，依次进栈，即依次插入元素1、2、3、4、5、A、B、C、D、E，依次出栈，即依次删除元素，根据栈“先进后出”的规则，应该以倒序出栈。即元素出栈顺序为EDCBA54321。

21、下列叙述中正确的是\_\_\_\_\_。

A. 循环队列中有队头和队尾两个指针，因此，循环队列是非线性结构

- B. 在循环队列中，只需要队头指针就能反映队列中元素的动态变化情况
- C. 在循环队列中，只需要队尾指针就能反映队列中元素的动态变化情况
- D. 循环队列中元素的个数是由队头指针和队尾指针共同决定

【答案】D

【解析】所谓循环队列，就是将队列存储空间的最后一个位置绕到第1个位置，形成逻辑上的环状空间，供队列循环使用。所以循环队列还是属于线性结构，所以选项A是错误的。循环队列的头指针front指向队列的第一个元素的前一位置，队尾指针rear指向队列的最后一个元素，循环队列的动态变化需要头尾指针共同反映，所以选项B、C是错误的。循环队列的长度是： $(sq.rear - sq.front + maxsize) \% maxsize$ ，所以循环队列的长度是由队头和队尾指针共同决定的，所以选项D正确。

22、在长度为n的有序线性表中进行二分查找，最坏情况下需要比较的次数是\_\_\_\_\_。

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

【答案】C

【解析】二分法检索要求线性表结点按关键值排序且以顺序方式存储。在查找时，首先与表的中间位置上结点的关键值比较，若相等则检索成功；否则根据比较结果确定下一步在表的前半部分或后半部分继续进行。二分法检索的效率比较高，设线性表有n个元素，则最多的检索次数为大于 $\log_2 n$  (2为底数)的最小整数，最少的检索次数为1。

23、下列叙述中正确的是\_\_\_\_\_。

- A. 顺序存储结构的存储一定是连续的，链式存储结构的存储空间不一定是连续的

- B. 顺序存储结构只针对线性结构，链式存储结构只针对非线性结构
- C. 顺序存储结构能存储有序表，链式存储结构不能存储有序表
- D. 链式存储结构比顺序存储结构节省存储空间

【答案】A

【解析】顺序存储结构就是用一组地址连续的存储单元依次存储该线性表中的各个元素，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，所以选项A是正确的。两者都可以存储线性的、有序的逻辑结构，所以选项B、C不正确。顺序结构使用的是连续物理空间，链式结构可以使用零散的物理空间存储，链式结构更灵活，不存在谁节约空间的说法，所以选项D不正确。故本题答案为A。

24、下列叙述中正确的是\_\_\_\_\_。

- A. 栈是"先进先出"的线性表
- B. 队列是"先进后出"的线性表
- C. 循环队列是非线性结构
- D. 有序线性表既可以采用顺序存储结构，也可以采用链式存储结构

【答案】D

【解析】栈是一种特殊的线性表，这种线性表只能在固定的一端进行插入和删除操作，允许插入和删除的一端称为栈顶，另一端称为栈底。一个新元素只能从栈顶一端进入，删除时，只能删除栈顶的元素，即刚刚被插入的元素。所以栈又称"后进先出"表，所以选项A是错误的。

队列可看作是插入在一端进行，删除在另一端进行的线性表，允许插入的一端称为队尾，允许删除的一端称为队头。在队列中，只能删除队头元素，队列的最后一个元素一定是最新入队的元素。因此队列又称"先进先出"表，所以选项B是错误的。

循环队列是将队列存储空间的最后一个位置绕到第一个位置，形成逻辑上的环状空间，供队

列循环使用。它的逻辑结构仍然是线性结构，所以选项C是错误的。

有序线性表既可以采用顺序存储结构，也可以采用链式存储结构，所以选项D是正确的。故本题答案为D。

25、支持子程序调用的数据结构是\_\_\_\_\_。

- A. 栈
- B. 树
- C. 队列
- D. 二叉树

【答案】A

【解析】栈是一种只能在一端进行插入或删除操作的线性表。在主程序调用子程序时要首先保存主程序当前的状态，然后去执行子程序，最终把子程序的执行结果返回到主程序调用子程序的位置，继续向下执行。这种调用符合栈"先进后出"的功能。

26、某二叉树有5个度为2的结点，则该二叉树中的叶子结点数是\_\_\_\_\_。

- A.10
- B.8
- C.6
- D.4

【答案】C

【解析】在任意一棵二叉树中，度为0的结点（即叶子结点）总是比度为2的结点多一个。本题中度为2的结点数为5，故叶子结点数为 $5+1=6$ 个。

27、下列排序方法中，最坏情况下比较次数最少的是\_\_\_\_\_。

- A. 冒泡排序

- B. 简单选择排序
- C. 直接插入排序
- D. 堆排序

【答案】D

【解析】(1) 冒泡排序法：是一种最简单的交换类排序法，它是通过相邻数据元素的交换逐步将线性表变成有序。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，需要比较的次数为 $n(n-1)/2$ 次。

(2) 简单插入排序法：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相同。在最坏情况下，简单插入排序需要 $n(n-1)/2$ 次比较。

(3) 简单选择排序法：对于长度为 $n$ 的序列，选择排序需要扫描 $n-1$ 遍，每一遍扫描均从剩下的子表中选出最小的元素，然后将该最小的元素与子表中的第一个元素进行交换。简单选择排序法在最坏情况下需要比较 $n(n-1)/2$ 次。

(4) 堆排序法：堆排序的方法为：①首先将一个无序序列建成堆。②然后将堆顶元素（序列中的最大项）与堆中最后一个元素交换（最大项应该在序列的最后）。在最坏情况下，堆排序需要比较的次数为 $O(n \log_2 n)$ 。假设线性表的长度为16，那么冒泡排序、直接插入排序、简单选择排序都需要比较120次，而堆排序需要比较64次。

28、下列数据结构中，属于非线性结构的是\_\_\_\_\_。

- A. 循环队列
- B. 带链队列
- C. 二叉树
- D. 带链栈

【答案】C

【解析】如果一个非空的数据结构满足下列两个条件：（1）有且只有一个根结点；（2）每一个结点最多有一个前件，也最多有一个后件。则称该数据结构为线性结构。本题中循环队列、带链队列和带链栈都是线性结构，二叉树是非线性结构。

29、下列数据结构中，能够按照"先进后出"原则存取数据的是\_\_\_\_\_。

- A. 循环队列
- B. 栈
- C. 队列
- D. 二叉树

【答案】B

【解析】在栈中，允许插入与删除的一端称为栈顶，而不允许插入与删除的另一端称为栈底。栈顶元素总是最后被插入的元素，从而也是最先能被删除的元素；栈底元素总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。

30、对于循环队列，下列叙述中正确的是\_\_\_\_\_。

- A. 队头指针是固定不变的
- B. 队头指针一定大于队尾指针
- C. 队头指针一定小于队尾指针
- D. 队头指针可以大于队尾指针，也可以小于队尾指针

【答案】D

【解析】所谓循环队列，就是将队列存储空间最后一个位置绕到第一个位置，形成逻辑上的环状空间，供队列循环使用。

在循环队列中，用队尾指针rear指向队列中的队尾元素，用排头指针front指向排头元素的前

一个位置。

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。当rear或front的值等于队列的长度+1时，就将rear或front的值置为1。一般情况下，rear大于front，因为入队的元素肯定比出队的元素多。特殊的情况是rear到达数组的上限之后又从数组的低端开始，此时，rear是小于front的。



高教考试在线