

数据结构和算法客观题库

1. 在数据结构中,从逻辑上可以把数据结构分为__C__。
 - A. 动态结构和静态结构
 - B. 紧凑结构和非紧凑结构
 - C. 线性结构和非线性结构
 - D. 内部结构和外部结构
2. 数据结构在计算机内存中的表示是指__A__。
 - A. 数据的存储结构
 - B. 数据结构
 - C. 数据的逻辑结构
 - D. 数据元素之间的关系
3. 在数据结构中,与所使用的计算机无关的是数据的__A__结构。
 - A. 逻辑
 - B. 存储
 - C. 逻辑和存储
 - D. 物理
4. 在存储数据时,通常不仅要存储各数据元素的值,而且还要存储__C__。
 - A. 数据的处理方法
 - B. 数据元素的类型
 - C. 数据元素之间的关系
 - D. 数据的存储方法
5. 在决定选取何种存储结构时,一般不考虑__A__。
 - A. 各结点的值如何
 - B. 结点个数的多少
 - C. 对数据有哪些运算
 - D. 所用的编程语言实现这种结构是否方便。
6. 以下说法正确的是__D__。
 - A. 数据项是数据的基本单位
 - B. 数据元素是数据的最小单位
 - C. 数据结构是带结构的数据项的集合
 - D. 一些表面上很不相同的数据可以有相同的逻辑结构
7. 算法分析的目的是__C__, 算法分析的两个主要方面是__A__。
 - (1) A. 找出数据结构的合理性
 - B. 研究算法中的输入和输出的关系
 - C. 分析算法的效率以求改进
 - D. 分析算法的易读性和文档性
 - (2) A. 空间复杂度和时间复杂度
 - B. 正确性和简明性
 - C. 可读性和文档性
 - D. 数据复杂性和程序复杂性
8. 下面程序段的时间复杂度是__ $O(n^2)$ __。


```
s=0;
for(i=0;i<n;i++)
    for(j=0;j<n;j++)
        s+=B[i][j];
sum=s;
```
9. 下面程序段的时间复杂度是__ $O(n*m)$ __。


```
for(i=0;i<n;i++)
    for(j=0;j<m;j++)
        A[i][j] = 0;
```
10. 下面程序段的时间复杂度是__ $O(\log_3 n)$ __。


```
i = 0;
while (i<=n)
    i = i * 3;
```

11. 在以下的叙述中, 正确的是 B。
- A. 线性表的顺序存储结构优于链表存储结构
B. 二维数组是其数据元素为线性表的线性表
C. 栈的操作方式是先进先出
D. 队列的操作方式是先进后出
12. 通常要求同一逻辑结构中的所有数据元素具有相同的特性, 这意味着 B。
- A. 数据元素具有同一特点
B. 不仅数据元素所包含的数据项的个数要相同, 而且对应的数据项的类型要一致
C. 每个数据元素都一样
D. 数据元素所包含的数据项的个数要相等
13. 链表不具备的特点是 A。
- A. 可随机访问任一结点 B. 插入删除不需要移动元素
C. 不必事先估计存储空间 D. 所需空间与其长度成正比
14. 不带头结点的单链表 head 为空的判定条件是 A。
- A. head == NULL B. head->next == NULL
C. head->next == head D. head != NULL
15. 带头结点的单链表 head 为空的判定条件是 B。
- A. head == NULL B. head->next == NULL
C. head->next == head D. head != NULL
16. 若某表最常用的操作是在最后一个结点之后插入一个结点或删除最后一个结点, 则采用 D 存储方式最节省运算时间。
- A. 单链表 B. 给出表头指针的单循环链表 C. 双链表 D. 带头结点的双循环链表
17. 需要分配较大空间, 插入和删除不需要移动元素的线性表, 其存储结构是 B。
- A. 单链表 B. 静态链表 C. 线性链表 D. 顺序存储结构
18. 非空的循环单链表 head 的尾结点 (由 p 所指向) 满足 C。
- A. p->next == NULL B. p == NULL
C. p->next == head D. p == head
19. 在循环双链表的 p 所指的结点之前插入 s 所指结点的操作是 D。
- A. p->prior = s; s->next = p; p->prior->next = s; s->prior = p->prior
B. p->prior = s; p->prior->next = s; s->next = p; s->prior = p->prior
C. s->next = p; s->prior = p->prior; p->prior = s; p->prior->next = s
D. s->next = p; s->prior = p->prior; p->prior->next = s; p->prior = s
20. 如果最常用的操作是取第 i 个结点及其前驱, 则采用 D 存储方式最节省时间。
- A. 单链表 B. 双链表 C. 单循环链表 D. 顺序表
21. 在一个具有 n 个结点的有序单链表中插入一个新结点并仍然保持有序的时间复杂度是 B。
- A. O(1) B. O(n) C. O(n²) D. O(nlog₂n)
22. 在一个长度为 n (n>1) 的单链表上, 设有头和尾两个指针, 执行 B 操作与链表的长度有关。
- A. 删除单链表中的第一个元素
B. 删除单链表中的最后一个元素

- C. 在单链表第一个元素前插入一个新元素
 D. 在单链表最后一个元素后插入一个新元素
23. 与单链表相比,双链表的优点之一是 D。
- A. 插入、删除操作更简单
 B. 可以进行随机访问
 C. 可以省略表头指针或表尾指针
 D. 顺序访问相邻结点更灵活
24. 如果对线性表的操作只有两种,即删除第一个元素,在最后一个元素的后面插入新元素,则最好使用 B。
- A. 只有表头指针没有表尾指针的循环单链表
 B. 只有表尾指针没有表头指针的循环单链表
 C. 非循环双链表
 D. 循环双链表
25. 在长度为 n 的顺序表的第 i 个位置上插入一个元素 ($1 \leq i \leq n+1$),元素的移动次数为: A。
- A. $n-i+1$ B. $n-i$ C. i D. $i-1$
26. 对于只在表的首、尾两端进行插入操作的线性表,宜采用的存储结构为 C。
- A. 顺序表 B. 用头指针表示的循环单链表
 C. 用尾指针表示的循环单链表 D. 单链表
27. 下述哪一条是顺序存储结构的优点? C。
- A 插入运算方便 B 可方便地用于各种逻辑结构的存储表示
 C 存储密度大 D 删除运算方便
28. 下面关于线性表的叙述中,错误的是哪一个? B。
- A 线性表采用顺序存储,必须占用一片连续的存储单元
 B 线性表采用顺序存储,便于进行插入和删除操作。
 C 线性表采用链式存储,不必占用一片连续的存储单元
 D 线性表采用链式存储,便于进行插入和删除操作。
29. 线性表是具有 n 个 B 的有限序列。
- A. 字符 B. 数据元素 C. 数据项 D. 表元素
30. 在 n 个结点的线性表的数组实现中,算法的时间复杂度是 $O(1)$ 的操作是 A。
- A. 访问第 i ($1 \leq i \leq n$) 个结点和求第 i 个结点的直接前驱 ($1 < i \leq n$)
 B. 在第 i ($1 \leq i \leq n$) 个结点后插入一个新结点
 C. 删除第 i ($1 \leq i \leq n$) 个结点
 D. 以上都不
31. 若长度为 n 的线性表采用顺序存储结构,在其第 i 个位置插入一个新元素的算法的时间复杂度为 C。
- A. $O(0)$ B. $O(1)$ C. $O(n)$ D. $O(n^2)$
32. 对于顺序存储的线性表,访问结点和增加、删除结点的时间复杂度为 C。
- A. $O(n)$ $O(n)$ B. $O(n)$ $O(1)$ C. $O(1)$ $O(n)$ D. $O(1)$ $O(1)$
33. 线性表 (a_1, a_2, \dots, a_n) 以链式方式存储,访问第 i 位置元素的时间复杂度为 C。
- A. $O(0)$ B. $O(1)$ C. $O(n)$ D. $O(n^2)$
34. 单链表中,增加一个头结点的目的是为了 C。

- A. 使单链表至少有一个结点 B. 标识表结点中首结点的位置
C. 方便运算的实现 D. 说明单链表是线性表的链式存储
35. 在单链表指针为 p 的结点之后插入指针为 s 的结点, 正确的操作是 B。
A. $p \rightarrow next = s; s \rightarrow next = p \rightarrow next$ B. $s \rightarrow next = p \rightarrow next; p \rightarrow next = s;$
C. $p \rightarrow next = s; p \rightarrow next = s \rightarrow next$ D. $p \rightarrow next = s \rightarrow next; p \rightarrow next = s$
36. 线性表的顺序存储结构是一种 A。
A. 随机存取的存储结构 B. 顺序存取的存储结构
C. 索引存取的存储结构 D. Hash 存取的存储结构
37. 栈的特点是 B, 队列的特点是 A。
A. 先进先出 B. 先进后出
38. 栈和队列的共同点是 C。
A. 都是先进后出 B. 都是先进先出
C. 只允许在端点处插入和删除元素 D. 没有共同点
39. 一个栈的进栈序列是 a, b, c, d, e , 则栈的不可能的输出序列是 C。
A. $edcba$ B. $decba$ C. $dceab$ D. $abcde$
40. 设有一个栈, 元素依次进栈的顺序为 A, B, C, D, E 。下列 C 是不可能的出栈序列。
A. A, B, C, D, E B. B, C, D, E, A C. E, A, B, C, D D. E, D, C, B, A
41. 以下 B 不是队列的基本运算?
A. 从队尾插入一个新元素 B. 从队列中删除第 i 个元素
C. 判断一个队列是否为空 D. 读取队头元素的值
42. 若已知一个栈的进栈序列是 $1, 2, 3, \dots, n$, 其输出序列为 $p_1, p_2, p_3, \dots, p_n$, 若 $p_1 = n$, 则 p_i 为 C。
A. i B. $n-i$ C. $n-i+1$ D. 不确定
43. 判定一个顺序栈 st (最多元素为 $MaxSize$) 为空的条件是 B。
A. $st \rightarrow top \neq -1$ B. $st \rightarrow top == -1$
C. $st \rightarrow top \neq MaxSize$ D. $st \rightarrow top == MaxSize$
44. 判定一个顺序栈 st (最多元素为 $MaxSize$) 为满的条件是 D。
A. $st \rightarrow top \neq -1$ B. $st \rightarrow top == -1$
C. $st \rightarrow top \neq MaxSize$ D. $st \rightarrow top == MaxSize$
45. 一个队列的入队序列是 $1, 2, 3, 4$, 则队列的输出序列是 B。
A. $4, 3, 2, 1$ B. $1, 2, 3, 4$
C. $1, 4, 3, 2$ D. $3, 2, 4, 1$
46. 判定一个循环队列 qu (最多元素为 $MaxSize$) 为空的条件是 C。
A. $qu \rightarrow rear - qu \rightarrow front == MaxSize$ B. $qu \rightarrow rear - qu \rightarrow front - 1 == MaxSize$
C. $qu \rightarrow rear == qu \rightarrow front$ D. $qu \rightarrow rear = qu \rightarrow front - 1$
47. 在循环队列中, 若 $front$ 与 $rear$ 分别表示队头元素和队尾元素的位置, 则判断循环队列空的条件是 C。
A. $front == rear + 1$ B. $rear == front + 1$ C. $front == rear$ D. $front == 0$
48. 向一个栈顶指针为 h 的带头结点的链栈中插入指针 s 所指的结点时, 应执行 D 操作。
A. $h \rightarrow next = s;$ B. $s \rightarrow next = h;$
C. $s \rightarrow next = h; h = s;$ D. $s \rightarrow next = h \rightarrow next; h \rightarrow next = s;$

49. 输入序列为 ABC, 可以变为 CBA 时, 经过的栈操作为 B。
- A. push, pop, push, pop, push, pop B. push, push, push, pop, pop, pop
C. push, push, pop, pop, push, pop D. push, pop, push, push, pop, pop
50. 若栈采用顺序存储方式存储, 现两栈共享空间 $V[1 \sim m]$, $top[1]$ 、 $top[2]$ 分别代表第 1 和第 2 个栈的栈顶, 栈 1 的底在 $V[1]$, 栈 2 的底在 $V[m]$, 则栈满的条件是 B。
- A. $|top[2]-top[1]|=0$ B. $top[1]+1=top[2]$ C. $top[1]+top[2]=m$
D. $top[1]=top[2]$
51. 设计一个判别表达式中左、右括号是否配对出现的算法, 采用 D 数据结构最佳。
- A. 线性表的顺序存储结构 B. 队列 C. 线性表的链式存储结构 D. 栈
52. 允许对队列进行的操作有 D。
- A. 对队列中的元素排序 B. 取出最近进队的元素
C. 在队头元素之前插入元素 D. 删除队头元素
53. 对于循环队列 D。
- A. 无法判断队列是否为空 B. 无法判断队列是否为满
C. 队列不可能满 D. 以上说法都不对
54. 若用一个大小为 6 的数值来实现循环队列, 且当前 $rear$ 和 $front$ 的值分别为 0 和 3, 当从队列中删除一个元素, 再加入两个元素后, $rear$ 和 $front$ 的值分别为 B。
- A. 1 和 5 B. 2 和 4 C. 4 和 2 D. 5 和 1
55. 队列的“先进先出”特性是指 D。
- A. 最早插入队列中的元素总是最后被删除
B. 当同时进行插入、删除操作时, 总是插入操作优先
C. 每当有删除操作时, 总是要先做一次插入操作
D. 每次从队列中删除的总是最早插入的元素
56. 和顺序栈相比, 链栈有一个比较明显的优势是 A。
- A. 通常不会出现栈满的情况 B. 通常不会出现栈空的情况
C. 插入操作更容易实现 D. 删除操作更容易实现
57. 用不带头结点的单链表存储队列, 其头指针指向队头结点, 尾指针指向队尾结点, 则在进行出队操作时 C。
- A. 仅修改队头指针 B. 仅修改队尾指针
C. 队头、队尾指针都可能要修改 D. 队头、队尾指针都要修改
58. 若串 $S='software'$, 其子串的数目是 B。
- A. 8 B. 37 C. 36 D. 9
59. 串的长度是指 B。
- A. 串中所含不同字母的个数 B. 串中所含字符的个数
C. 串中所含不同字符的个数 D. 串中所含非空格字符的个数
60. 串是一种特殊的线性表, 其特殊性体现在 B。
- A. 可以顺序存储 B. 数据元素是一个字符
C. 可以链式存储 D. 数据元素可以是多个字符
61. 设有两个串 p 和 q , 求 q 在 p 中首次出现的位置的运算称为 B。
- A. 连接 B. 模式匹配 C. 求子串 D. 求串长
62. 数组 A 中, 每个元素的长度为 3 个字节, 行下标 i 从 1 到 8, 列下标 j 从 1 到

10. 从首地址 SA 开始连续存放的存储器内, 该数组按行存放, 元素 A[8][5]的起始地址为 C。

A. SA+141 B. SA+144 C. SA+222 D. SA+225

63. 数组 A 中, 每个元素的长度为 3 个字节, 行下标 i 从 1 到 8, 列下标 j 从 1 到 10, 从首地址 SA 开始连续存放的存储器内, 该数组按行存放, 元素 A[5][8]的起始地址为 C。

A. SA+141 B. SA+180 C. SA+222 D. SA+225

64. 若声明一个浮点数组如下: float average[]=new float[30];

假设该数组的内存起始位置为 200, average[15]的内存地址是 C。

A. 214 B. 215 C. 260 D. 256

65. 设二维数组 A[1... m, 1... n]按行存储在数组 B 中, 则二维数组元素 A[i,j]在一维数组 B 中的下标为 A。

A. n*(i-1)+j B. n*(i-1)+j-1 C. i*(j-1) D. j*m+i-1

66. 有一个 100×90 的稀疏矩阵, 非 0 元素有 10, 设每个整型数占 2 个字节, 则用三元组表示该矩阵时, 所需的字节数是 B。

A. 20 B. 66 C. 18 000 D. 33

67. 数组 A[0 ... 4, -1 ... -3, 5 ... 7]中含有的元素个数是 A。

A. 55 B. 45 C. 36 D. 16

68. 对矩阵进行压缩存储是为了 D。

A. 方便运算 B. 方便存储 C. 提高运算速度 D. 减少存储空间

69. 设有一个 10 阶的对称矩阵 A, 采用压缩存储方式, 以行序为主存储, a_{1,1} 为第一个元素, 其存储地址为 1, 每个元素占 1 个地址空间, 则 a_{8,5} 的地址为 B。

A. 13 B. 33 C. 18 D. 40

70. 稀疏矩阵一般的压缩存储方式有两种, 即 C。

A. 二维数组和三维数组 B. 三元组和散列
C. 三元组和十字链表 D. 散列和十字链表

71. 树最适合用来表示 C。

A. 有序数据元素 B. 无序数据元素
C. 元素之间具有分支层次关系的数据 D. 元素之间无联系的数据

72. 深度为 5 的二叉树至多有 C 个结点。

A. 16 B. 32 C. 31 D. 10

73. 对一个满二叉树, m 个叶子, n 个结点, 深度为 h, 则 D。

A. n = h+m B. h+m = 2n C. m = h-1 D. n = 2^h-1

74. 任何一棵二叉树的叶子结点在前序、中序和后序遍历序列中的相对次序 A。

A. 不发生改变 B. 发生改变 C. 不能确定 D. 以上都不对

75. 在线索化树中, 每个结点必须设置一个标志来说明它的左、右链指向的是树结构信息, 还是线索化信息, 若 0 标识树结构信息, 1 标识线索, 对应叶结点的左右链域, 应标识为 D。

A. 00 B. 01 C. 10 D. 11

76. 在下述论述中, 正确的是 D。

①只有一个结点的二叉树的度为 0; ②二叉树的度为 2; ③二叉树的左右子树可任意交换;

④深度为 K 的顺序二叉树的结点个数小于或等于深度相同的满二叉树。

- A. ①②③ B. ②③④ C. ②④ D. ①④

77. 设森林 F 对应的二叉树为 B, 它有 m 个结点, B 的根为 p, p 的右子树的结点个数为 n, 森林 F 中第一棵树的结点的个数是 A。

- A. m-n B. m-n-1 C. n+1 D. 不能确定

78. 若一棵二叉树具有 10 个度为 2 的结点, 5 个度为 1 的结点, 则度为 0 的结点的个数是 B。

- A. 9 B. 11 C. 15 D. 不能确定

79. 具有 10 个叶子结点的二叉树中有 B 个度为 2 的结点。

- A. 8 B. 9 C. 10 D. 11

80. 在一个无向图中, 所有顶点的度数之和等于所有边数的 C 倍。

- A. 1/2 B. 1 C. 2 D. 4

81. 在一个有向图中, 所有顶点的入度之和等于所有顶点的出度之和的 B 倍。

- A. 1/2 B. 1 C. 2 D. 4

82. 某二叉树结点的中序序列为 ABCDEFG, 后序序列为 BDCAFGE, 则其左子树中结点数目为: C

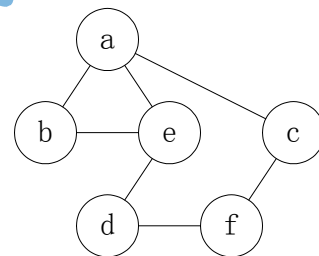
- A. 3 B. 2 C. 4 D. 5

83. 已知一算术表达式的中缀形式为 $A+B * C-D/E$, 后缀形式为 $ABC * +DE / -$, 其前缀形式为 D。

- A. $-A+B * C/DE$ B. $-A+B * CD/E$ C. $-+ * ABC/DE$ D. $-+A * BC/DE$

84. 已知一个图, 如图所示, 若从顶点 a 出发按深度搜索法进行遍历, 则可能得到的一种顶点序列为 D; 按广度搜索法进行遍历, 则可能得到的一种顶点序列为 A;

- ① A. a, b, e, c, d, f B. a, c, f, e, b, d
C. a, e, b, c, f, d D. a, e, d, f, c, b
② A. a, b, c, e, d, f B. a, b, c, e, f, d
C. a, e, b, c, f, d D. a, c, f, d, e, b



85. 采用邻接表存储的图的深度优先遍历算法类似于二叉树的 A。

- A. 先序遍历 B. 中序遍历 C. 后序遍历 D. 按层遍历

86. 采用邻接表存储的图的广度优先遍历算法类似于二叉树的 D。

- A. 先序遍历 B. 中序遍历 C. 后序遍历 D. 按层遍历

87. 具有 n 个结点的连通图至少有 A 条边。

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

88. 广义表 $((a), a)$ 的表头是 C, 表尾是 C。

- A. a B. () C. (a) D. ((a))

89. 广义表 $((a))$ 的表头是 C, 表尾是 B。

- A. a B. () C. (a) D. ((a))

90. 顺序查找法适合于存储结构为 B 的线性表。

- A. 散列存储 B. 顺序存储或链式存储 C. 压缩存储 D. 索引存储

91. 对线性表进行折半查找时, 要求线性表必须 B。

- A. 以顺序方式存储 B. 以顺序方式存储, 且结点按关键字有序排列
C. 以链式方式存储 D. 以链式方式存储, 且结点按关键字有序排列

92. 采用折半查找法查找长度为 n 的线性表时, 每个元素的平均查找长度为 D。

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

93. 有一个有序表为 {1, 3, 9, 12, 32, 41, 45, 62, 75, 77, 82, 95, 100}, 当

折半查找值为 82 的结点时, C 次比较后查找成功。

- A. 11 B. 5 C. 4 D. 8

94. 二叉树为二叉排序树的充分必要条件是任一结点的值均大于其左孩子的值、小于其右孩子的值。这种说法 B。

- A. 正确 B. 错误

95. 下面关于 B 树和 B+树的叙述中, 不正确的结论是 A。

- A. B 树和 B+树都能有效的支持顺序查找 B. B 树和 B+树都能有效的支持随机查找
C. B 树和 B+树都是平衡的多叉树 D. B 树和 B+树都可用于文件索引结构

96. 以下说法错误的是 B。

- A. 散列法存储的思想是由关键字值决定数据的存储地址
B. 散列表的结点中只包含数据元素自身的信息, 不包含指针。
C. 负载因子是散列表的一个重要参数, 它反映了散列表的饱满程度。
D. 散列表的查找效率主要取决于散列表构造时选取的散列函数和处理冲突的方法。

97. 查找效率最高的二叉排序树是 C。

- A. 所有结点的左子树都为空的二叉排序树。
B. 所有结点的右子树都为空的二叉排序树。
C. 平衡二叉树。
D. 没有左子树的二叉排序树。

98. 排序方法中, 从未排序序列中依次取出元素与已排序序列中的元素进行比较, 将其放入已排序序列的正确位置上的方法, 称为 C。

- A. 希尔排序 B. 冒泡排序 C. 插入排序 D. 选择排

99. 在所有的排序方法中, 关键字比较的次数与记录的初始排列次序无关的是 D。

- A. 希尔排序 B. 冒泡排序 C. 直接插入排序 D. 直接选择排序

100. 堆是一种有用的数据结构。下列关键码序列 D 是一个堆。

- A. 94,31,53,23,16,72 B. 94,53,31,72,16,23
C. 16,53,23,94,31,72 D. 16,31,23,94,53,72

101. 堆排序是一种 B 排序。

- A. 插入 B. 选择 C. 交换 D. 归并

102. D 在链表中进行操作比在顺序表中进行操作效率高。

- A. 顺序查找 B. 折半查找 C. 分块查找 D. 插入

103. 直接选择排序的时间复杂度为 D。(n 为元素个数)

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