

1.已知一个文件中出现的各字符及其对应的频率如下表所示。采用 Huffman 编码,则该文件中字符 a 和 c 的码长分别为()。若采用 Huffman 编码,则字序列“110001001101”的编码应为()。

字符	a	b	c	d	e	f
频率(%)	45	13	12	16	9	5

问题 1: A.1 和 3 B.1 和 4 C.3 和 3 D.3 和 4

问题 2: A.face B.bace C.acde D.fade

2.归并排序算法在排序过程中,将待排序数组分为两个大小相同的子数组,分别对两个子数组采用归并排序算法进行排序,排好序的两个子数组采用时间复杂度为 $\Theta(n)$ 的过程合并为一个大数组。根据上述描述,归并排序算法采用了()算法设计策略。归并排序算法的最好和最坏情况下的时间复杂度为()。

问题 1: A.分治 B.动态规划 C.贪心 D.回溯

问题 2: A. $\Theta(n)$ 和 $\Theta(\lg n)$ B. $\Theta(n)$ 和 $\Theta(n^2)$ C. $\Theta(\lg n)$ 和 $\Theta(\lg n)$ D. $\Theta(\lg n)$ 和 $\Theta(n^2)$

3.对有向图 G 进行拓扑排序得到的拓扑序列中,顶点 V_i 在顶点 V_j 之前,则说明 G 中()

A.一定存在有向弧 $\langle V_i, V_j \rangle$ B.一定不存在有向弧 $\langle V_j, V_i \rangle$

C.必定存在从 V_i 到 V_j 的路径 D.必定存在从 V_j 到 V_i 的路径

4.n 个关键码构成的序列 $\{k_1, k_2, \dots, k_n\}$ 当且仅当满足下列关系时称其为堆。

$$\begin{cases} k_i \leq k_{2i} \\ k_i \leq k_{2i+1} \end{cases} \text{ 或 } \begin{cases} k_i \geq k_{2i} \\ k_i \geq k_{2i+1} \end{cases}$$

以下关键码序列中, () 不是堆。

A.15, 25, 21, 53, 73, 65, 33

B.15, 25, 21, 33, 73, 65, 53

C.73, 65, 25, 21, 15, 53, 33

D.73, 65, 25, 33, 53, 15, 21

5.二叉树的高度是指其层数,空二叉树的高度为 0,仅有根结点的二叉树高度为 1,若某二叉树中共有 1024 个结点,则该二叉树的高度是整数区间()中的任一值。

A.(10, 1024)

B.[10, 1024]

C.(11, 1024)

D.[11, 1024]

6.采用循环队列的优点是()

A.入队和出队可以在队列的同端点进行操作

B.入队和出队操作都不需要移动队列中的其他元素

C.避免出现队列满的情况

D.避免出现队列空的情况

7.若栈采用顺序存储方式,现有两栈共享空间 $V[1..n]$, $\text{top}[i]$ 代表 i ($i=1, 2$) 个栈的栈顶(两个栈都空时 $\text{top}[1]=1$ 、 $\text{top}[2]=n$), 栈 1 的底在 $V[1]$, 栈 2 的底在 $V[n]$, 则栈满(即 n 个元素暂存在这两个栈)的条件是()

A. $\text{top}[1]=\text{top}[2]$

B. $\text{top}[1]+\text{top}[2]==1$

C. $\text{top}[1]+\text{top}[2]==n$

D. $\text{top}[1]-\text{top}[2]==1$

8.在求解某问题时,经过分析发现该问题具有最优子结构和重叠子问题性质。则适用()算法设计策略得到最优解。若了解问题的解空间,并以广度优先的方式搜索解空间,则采用的是()算法策略。

问题 1: A.分治

B.贪心

C.动态规则

D.回溯

问题 2: A.动态规则

B.贪心

C.回溯

D.分支限界

9.最大尺寸和问题描述为，在 n 个整数（包含负数）的数组 A 中，求之和最大的非空连续子数组，如数组 $A = (-2, 11, -4, 13, -5, -2)$ ，其中子数组 $B = (11, -4, 13)$ 具有最大子段和 20 ($11-4+13=20$)。求解该问题时，可以将数组分为两个 $n/2$ 个整数的子数组最大子段或或者在前半段，或者在后半段，或者跨越中间元素，通过该方法继续划分问题，直至最后求出最大子段和，该算法的时间复杂度为（ ）。

- A. $O(n \lg n)$ B. $O(n^2)$ C. $O(n^2 \lg n)$ D. $O(n^3)$

10.对数组 $A = (2, 8, 7, 1, 3, 5, 6, 4)$ 构建大顶堆为（ ）（用数组表示）

- A. (1, 2, 3, 4, 5, 6, 7, 8) B. (1, 2, 5, 4, 3, 7, 6, 8)
C. (8, 4, 7, 2, 3, 5, 6, 1) D. (8, 7, 6, 5, 4, 3, 2, 1)

11.对于一个初始无序的关键字序列，在下面的排序方法中，（ ）第一趟排序结束后，一定能将序列中的某个元素在最终有序序列中的位置确定下来。

①直接插入排序 ②冒泡排序 ③简单选择排序 ④堆排序 ⑤快速排序 ⑥归并排序

- A. ①②③⑥ B. ①②③⑤⑥ C. ②③④⑤ D. ③④⑤⑥

12.设用线性探查法解决冲突构造哈希表，且哈希函数为 $H(key) = key \% m$ ，若在该哈希表中查找某关键字 e 是成功的且与多个关键字进行了比较，则（ ）。

- A. 这些关键字形成一个有序序列
B. 这些关键字都不是 e 的同义词
C. 这些关键字都是 e 的同义词
D. 这些关键字的第一个可以不是 e 的同义词

13.（ ）是对稀疏矩阵进行压缩存储的方式。

- A. 二维数组和双向链表 B. 三元组顺序表和十字链表
C. 邻接矩阵和十字链表 D. 索引顺序表和双向链表

14.当二叉数中的结点数确定时，（ ）的高度一定是最小的。

- A. 二叉排序数 B. 完全二叉树 C. 线索二叉树 D. 最优二叉树

15.设有栈 S 和队列 Q 初始状态为空，数据元素序列 a, b, c, d, e, f 依次通过栈 S ，且多个元素从 S 出栈后立即进入队列 Q ，若出队的序列是 b, d, f, e, c, a ，则 S 中的元素最多时，栈底到栈顶的元素依次为（ ）。

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

16.根据渐进分析，表达式序列： $n^4, \lg n, 2^n, 1000n, n^{2/3}, n!$ 从低到高排序为（ ）。

- A. $\lg n, 1000n, n^{2/3}, n^4, n!, 2^n$ B. $n^{2/3}, 1000n, \lg n, n^4, n!, 2^n$
C. $\lg n, 1000n, n^{2/3}, 2^n, n^4, n!$ D. $\lg n, n^{2/3}, 1000n, \lg n, n^4, 2^n, n!$

17.某简单无向连通图 G 的顶点数为 n ，则图 G 最少和最多分别有（ ）条边。

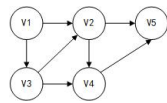
- A. $n, n^2/2$ B. $n-1, n*(n-1)/2$ C. $n, n*(n-1)/2$ D. $n-1, n^2/2$

18.对数组 $A = (2, 8, 7, 1, 3, 5, 6, 4)$ 用快速排序算法的划分方法进行一趟划分后得到的数组 A 为（ ）（非递减排序，以最后一个元素为基准元素）。进行一趟划分的计算时间为（ ）。

问题 1: A. (1, 2, 8, 7, 3, 5, 6, 4) B. (1, 2, 3, 4, 8, 7, 5, 6) C. (2, 3, 1, 4, 7, 5, 6, 8) D. (2, 1, 3, 4, 8, 7, 5, 6)

问题 2: A. $O(1)$ B. $O(\lg n)$ C. $O(n)$ D. $O(n \lg n)$

19.某有向图如下所示，从顶点 v1 出发对其进行深度优先遍历，可能得到的遍历序列是（ ）；从顶点 v1 出发对其进行广度优先遍历，可能得到的遍历序列是（ ）。



①v1 v2 v3 v4 v5

②v1 v3 v4 v5 v2

③v1 v3 v2 v4 v5

④v1 v2 v4 v5 v3

问题 1: A. ①②③ B. ①③④ C. ①②④ D. ②③④

问题 2: A. ①② B. ①③ C. ②③ D. ③④

20.在线性表 L 中进行二分查找，要求 L（ ）。

A. 顺序存储，元素随机排列 B. 双向链表存储，元素随机排列

C. 顺序存储，元素有序排列 D. 双向链表存储，元素有序排列

21.通过元素在存储空间中的相对位置来表示数据元素之间的逻辑关系，是（ ）的特点。

A. 顺序存储 B. 链表存储 C. 索引存储 D. 哈希存储

22.以下关于 Huffman (哈夫曼)树的叙述中，错误的是（ ）。

A. 权值越大的叶子离根结点越近

B. Huffman (哈夫曼)树中不存在只有一个子树的结点

C. Huffman (哈夫曼)树中的结点总数一定为奇数

D. 权值相同的结点到树根的路径长度一定相同

23.已知某文档包含 5 个字符，每个字符出现的频率如下表所示。采用霍夫曼编码对该文档压缩存储，则单词"cade"的编码为（ ），文档的压缩比为（ ）。

字符	a	b	c	d	e
频率 (%)	40	10	20	16	14

问题 1: A. 1110110101 B. 1100111101 C. 1110110100 D. 1100111100

问题 2: A. 20% B. 25% C. 27% D. 30%

24.采用贪心算法保证能求得最优解的问题是（ ）。

A. 0-1 背包 B. 矩阵链乘 C. 最长公共子序列 D. 部分（分数）背包

25.对 n 个数排序，最坏情况下时间复杂度最低的算法是（ ）排序算法。

A. 插入 B. 冒泡 C. 归并 D. 快速

26.对于如下所示的有向图，其邻接矩阵是一个（ ）的矩阵。采用邻接链表存储时，顶点 1 的表结点个数为 2，顶点 5 的表结点个数为 0，顶点 2 和 3 的表结点个数分别为（ ）。

问题 1: A. 5×5 B. 5×7 C. 7×5 D. 7×7

问题 2: A. 2.1 B. 2.2 C. 3.4 D. 4.3

27.某树共有 n 个结点，其中所有分支结点的度为 k （即每个非叶子结点的子树数目），则该树中叶子结点的个数为（ ）。

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

28.某二叉树的中序、先序遍历序列分别为{20, 30, 10, 50, 40}、{10, 20, 30, 40, 50}，则该二叉树的后序遍历序列为（ ）。

- A. 50, 40, 30, 20, 10 B. 30, 20, 10, 50, 40
C. 30, 20, 50, 40, 10 D. 20, 30, 10, 40, 50

29.对于一个 n 阶的对称矩阵 A ，将其下三角区域（含主对角线）的元素按行存储在一维数组 S 中，设元素 $A[i][j]$ 存放在 $S[k]$ 中，且 $S[1]=A[0][0]$ ，则 k 与 i, j ($i \leq j$) 的对应关系是（ ）。

- A. $k = i(i+1)/2 + j - 1$ B. $k = j(j+1)/2 + i - 1$
C. $k = i(i+1)/2 + j + 1$ D. $k = j(j+1)/2 + i - 1$

30.对于有序表(8,15,19,23,26,31,40,65,91)，用二分法进行查找时，可能的关键字比较顺序为（ ）。

- A. 26, 23, 19 B. 26, 8, 19 C. 26, 40, 65 D. 26, 31, 40

31.已知矩阵 $A_m \times n$ 和 $B_n \times p$ 相乘的时间复杂度为 $O(mnp)$ 。矩阵相乘满足结合律，如三个矩阵 A 、 B 、 C 相乘的顺序可以是 $(A*B)*C$ 也可以是 $A*(B*C)$ 。不同的相乘顺序所需进行的乘法次数可能有很大的差别。因此确定 n 个矩阵相乘的最优计算顺序是一个非常重要的问题。已知确定 n 个矩阵 $A_1 A_2 \dots A_n$ 相乘的计算顺序具有最优子结构，即 $A_1 A_2 \dots A_n$ 的最优计算顺序包含其子问题 $A_1 A_2 \dots A_k$ 和 $A_{k+1} A_{k+2} \dots A_n$ ($1 \leq k < n$) 的最优计算顺序。

可以列出其递归式为：

$$m[i, j] = \begin{cases} 0 & \text{if } i=j \\ \min_{i \leq k < j} \{m[i, k] + m[k+1, j] + P_{i-1} P_k P_j\} & \text{if } i < j \end{cases}$$

其中， A_i 的维度为 $P_{i-1} \times P_i$, $m[i, j]$ 表示 $A_i A_{i+1} \dots A_j$ 最优计算顺序的相乘次数。先采用自底向上的方法求 n 个矩阵相乘的最优计算顺序。则求解该问题的算法设计策略为（ ）。算法的时间复杂度为（ ），空间复杂度为（ ）。给定一个实例， $(p_0 p_1 \dots p_5) = (20, 15, 4, 10, 20, 25)$ ，最优计算顺序为（ ）。

- 问题 1: A. 分治法 B. 动态规划法 C. 贪心法 D. 回溯法
问题 2: A. $O(n^2)$ B. $O(n^2 \lg n)$ C. $O(n^3)$ D. $O(2^n)$
问题 3: A. $O(n^2)$ B. $O(n^2 \lg n)$ C. $O(n^3)$ D. $O(2^n)$
问题 4:

- A. $((A_1 \times A_2) \times A_3) \times A_4 \times A_5$ B. $A_1 \times (A_2 \times (A_3 \times (A_4 \times A_5)))$
C. $((A_1 \times A_2) \times A_3) \times (A_4 \times A_5)$ D. $(A_1 \times A_2) \times ((A_3 \times A_4) \times A_5)$

32.设散列函数为 $H(\text{key}) = \text{key} \% 11$ ，对于关键码序列(23, 40, 91, 17, 19, 10, 31, 65, 26)，用线性探查法解决冲突构造的哈希表为（ ）。

A.

哈希地址	0	1	2	3	4	5	6	7	8	9	10
关键码	10	23		91	26		17	40	19	31	65

B.

哈希地址	0	1	2	3	4	5	6	7	8	9	10
关键码	65	23		91	26		17	40	19	31	10

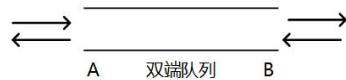
C.

哈希地址	0	1	2	3	4	5	6	7	8	9	10
关键码		23	10	91	26		17	40	19	31	65

D.

哈希地址	0	1	2	3	4	5	6	7	8	9	10
关键码		23	65	91	26		17	40	19	31	10

33.双端队列是指在队列的两个端口都可以加入和删除元素，如下图所示。现在要求元素进队列和出队列必须在同一端口，即从A端进队的元素必须从A端出、从B端进队的元素必须从B端出，则对于4个元素的序列a、b、c、d，若要求前2个元素(a、b)从A端口按次序全部进入队列，后两个元素(c、d)从B端口按次序全部进入队列，则不可能得到的出队序列是()。



- A. d、a、b、c B. d、c、b、a
C. b、a、d、c D. b、d、c、a

34.具有3个结点的二叉树有5种，可推测出具有4个结点的二叉树有()种。

- A.10 B.11 C.14 D.15

35.某n阶的三对角矩阵A如下图所示，按行将元素存储在一维数组M中，设 $a_{1,1}$ 存储在M[1]，那么 $a_{i,j}$ ($1 \leq i, j \leq n$ 且 $a_{i,j}$ 位于三条对角线中)存储在M()。

$$A_{n \times n} = \begin{pmatrix} a_{1,1} & a_{1,2} & & & & \\ a_{2,1} & a_{2,2} & a_{2,3} & & & \\ & a_{3,2} & a_{3,3} & a_{3,4} & & \\ & & \dots & \dots & \dots & \\ & & & a_{i,j-1} & a_{i,j} & a_{i,j+1} \\ & 0 & & & \dots & \dots \\ & & & & & a_{n,n-1} & a_{n,n} \end{pmatrix}$$

- A. $i+2j$ B. $2i+j$ C. $i+2j-2$ D. $2i+j-2$

36.在一条笔直公路的一边有许多房子，现要安装消防栓，每个消防栓的覆盖范围远大于房子的面积，如下图所示。现求解能覆盖所有房子的最少消防栓数和安装方案(问题求解过程中，可将房子和消防栓均视为直线上的点)。



该问题求解算法的基本思路为：从左端的第一栋房子开始，在其右侧m米处安装一个消防栓，去掉被该消防栓覆盖的所有房子。在剩余的房子中重复上述操作，直到所有房子被覆盖。算法采用的设计策略为()；对应的时间复杂度为()。假设公路起点A的坐标为0，消防栓的覆盖范围(半径)为20米，10栋房子的坐标为(10, 20, 30, 35, 60, 80, 160, 210, 260, 300)，单位为米。根据上述算法，共需要安装()个消防栓。以下关于该求解算法的叙述中，正确的是()。

- 问题1: A.分治 B.动态规划 C.贪心 D.回溯
问题2: A. $\theta(\lg n)$ B. $\theta(n)$ C. $\theta(n \lg n)$ D. $\theta(n^2)$
问题3: A.4 B.5 C.6 D.7

问题4:

- A.肯定可以求得问题的一个最优解
B.可以求得问题的所有最优解
C.对有些实例，可能得不到最优解
D.只能得到近似最优解

37.图 G 的邻接矩阵如下图所示（顶点依次表示为 v_0 、 v_1 、 v_2 、 v_3 、 v_4 、 v_5 ），G 是（ ）。对 G 进行广度优先遍历（从 v_0 开始），可能的遍历序列为（ ）。

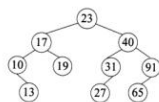
∞	18	17	∞	∞	∞
∞	∞	∞	20	16	∞
∞	19	∞	23	∞	∞
∞	∞	∞	∞	∞	15
∞	∞	∞	∞	∞	12
∞	∞	∞	∞	∞	∞

问题 1: A. 无向图 B. 有向图 C. 完全图 D. 强连通图

问题 2:

- A. v_0 、 v_1 、 v_2 、 v_3 、 v_4 、 v_5 B. v_0 、 v_2 、 v_4 、 v_5 、 v_1 、 v_3
 C. v_0 、 v_1 、 v_3 、 v_5 、 v_2 、 v_4 D. v_0 、 v_2 、 v_4 、 v_3 、 v_5 、 v_1

38.可以构造出下图所示二叉排序树（二叉检索树、二叉查找树）的关键码序列是（ ）。



- A. 10 13 17 19 23 27 31 40 65 91 B. 23 40 91 17 19 10 31 65 27 13
 C. 23 19 40 27 17 13 10 91 65 31 D. 27 31 40 65 91 13 10 17 23 19

39.已知某二叉树的先序遍历序列为 A B C D E F、中序遍历序列为 B A D C F E，则可以确定该二叉树（ ）。

- A. 是单支树（即非叶子结点都只有一个孩子） B. 高度为 4（即结点分布在 4 层上）
 C. 根结点的左子树为空 D. 根结点的右子树为空

40.栈的特点是后进先出，若用单链表作为栈的存储结构，并用头指针作为栈顶指针，则（ ）。

- A. 入栈和出栈操作都不需要遍历链表 B. 入栈和出栈操作都需要遍历链表
 C. 入栈操作需要遍历链表而出栈操作不需要 D. 入栈操作不需要遍历链表而出栈操作需要

41.现需要申请一些场地举办一批活动，每个活动有开始时间和结束时间。在同一个场地，如果一个活动结束之前，另一个活动开始，即两个活动冲突。若活动 A 从 1 时间开始，5 时间结束，活动 B 从 5 时间开始，8 时间结束，则活动 A 和 B 不冲突。现要计算 n 个活动需要的最少场地数。

求解该问题的基本思路如下（假设需要场地数为 m ，活动数为 n ，场地集合为 P_1, P_2, \dots, P_m ），初始条件 P_i 均无活动安排：

- （1）采用快速排序算法对 n 个活动的开始时间从小到大排序，得到活动 a_1, a_2, \dots, a_n 。对每个活动 a_i ， i 从 1 到 n ，重复步骤（2）、（3）和（4）；
- （2）从 p_1 开始，判断 a_i 与 P_1 的最后一个活动是否冲突，若冲突，考虑下一个场地 p_2, \dots ；
- （3）一旦发现 a_i 与某个 p_j 的最后一个活动不冲突，则将 a_i 安排到 P_j ，考虑下一个活动；
- （4）若 a_i 与所有已安排活动的 p_j 的最后一个活动均冲突，则将 a_i 安排到一个新的场地，考虑下一个活动；
- （5）将 n 减去没有安排活动的场地数即可得到所用的最少场地数。

算法首先采用了快速排序算法进行排序，其算法设计策略是（ ）；后面步骤采用的算法设计策略是（ ）。整个算法的时间复杂度是（ ）。下表给出了 $n=11$ 的活动集合，根据上述算法，得到最少的场地数为（ ）。

i	1	2	3	4	5	6	7	8	9	10	11
开始时间 s_i	0	1	2	3	3	5	5	6	8	8	12
结束时间 t_i	6	4	13	5	8	7	9	10	11	12	14

问题 1: A. 分治 B. 动态规划 C. 贪心 D. 回溯 问题 2: A. 分治 B. 动态规划 C. 贪心 D. 回溯

问题 3: A. $\theta(\lg n)$ B. $\theta(n)$ C. $\theta(n \lg n)$ D. $\theta(n^2)$ 问题 4: A. 4 B. 5 C. 6 D. 7

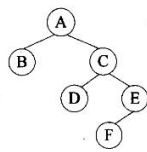
42.对有 n 个结点、 e 条边且采用数组表示法（即邻接矩阵存储）的无向图进行深度优先遍历，时间复杂度为（ ）。

- A. $O(n^2)$ B. $O(e^2)$ C. $O(n+e)$ D. $O(n*e)$

43.用哈希表存储元素时，需要进行冲突（碰撞）处理，冲突是指（ ）。

- A. 关键字被依次映射到地址编号连续的存储位置
B. 关键字不同的元素被映射到相同的存储位置
C. 关键字相同的元素被映射到不同的存储位置
D. 关键字被映射到哈希表之外的位置

44.对下面的二叉树进行顺序存储（用数组 MEM 表示），已知结点 A、B、C 在 MEM 中对应元素的下标分别为 1、2、3，那么结点 D、E、F 对应的数组元素下标为（ ）。



- A. 4、5、6 B. 4、7、10 C. 6、7、8 D. 6、7、14

45.设有 n 阶三对角矩阵 A ，即非零元素都位于主对角线以及与主对角线平行且紧邻的两条对角线上，现对该矩阵进行按行压缩存储，若其压缩空间用数组 B 表示， A 的元素下标从 0 开始， B 的元素下标从 1 开始。已知 $A[0, 0]$ 存储在 $B[1]$ ， $A[n-1, n-1]$ 存储在 $B[3n-2]$ ，那么非零元素 $A[i, j]$ ($0 \leq i < n, 0 \leq j < n, |i-j| \leq 1$) 存储在 $B[\quad]$ 。

- A. $2i+j-1$ B. $2i+j$ C. $2i+j+1$ D. $3i-j+1$

46.队列的特点是先进先出，若用循环单链表表示队列，则（ ）。

- A. 入队列和出队列操作都不需要遍历链表
B. 入队列和出队列操作都需要遍历链表
C. 入队列操作需要遍历链表而出队列操作不需要
D. 入队列操作不需要遍历链表而出队列操作需要

47.现需要对一个基本有序的数组进行排序。此时最适宜采用的算法为（ ）排序算法，时间复杂度为（ ）。

- 问题 1: A. 插入 B. 快速 C. 归并 D. 堆

- 问题 2: A. $O(n)$ B. $O(n \lg n)$ C. $O(n^2)$ D. $O(n^2 \lg n)$

48.求解两个长度为 n 的序列 X 和 Y 的一个最长公共子序列（如序列 ABCBDAB 和 BDCABA 的一个最长公共子序列为 BCBA）可以采用多种计算方法。如可以采用蛮力法，对 X 的每一个子序列，判断其是否也是 Y 的子序列，最后求出最长的即可，该方法的时间复杂度为（ ）。经分析发现该问题具有最优子结构，可以定义序列长度分别为 i 和 j 的两个序列 X 和 Y 的最长公共子序列的长度为 $C[i, j]$ ，如下式所示。

$$C[i, j] = \begin{cases} 0 & \text{若 } i = 0 \text{ 或 } j = 0 \\ C[i-1, j-1] + 1 & \text{若 } i, j > 0 \text{ 且 } x_i = y_j \\ \max(C[i-1, j], C[i, j-1]) & \text{其他} \end{cases}$$

采用自底向上的方法实现该算法，则时间复杂度为（ ）。

- 问题 1: A. $O(n^2)$ B. $O(n^2 \lg n)$ C. $O(n^3)$ D. $O(n2^n)$
问题 2: A. $O(n^2)$ B. $O(n^2 \lg n)$ C. $O(n^3)$ D. $O(n2^n)$

49.两个递增序列 A 和 B 的长度分别为 m 和 n ($m < n$ 且 m 与 n 接近), 将二者归并为一个长度为 m+n 的递增序列。当关系为 () 时, 归并过程中元素的比较次数最少。

- A. $a_1 < a_2 < \dots < a_{m-1} < a_m < b_1 < b_2 < \dots < b_{n-1} < b_n$
- B. $b_1 < b_2 < \dots < b_{n-1} < b_n < a_1 < a_2 < \dots < a_{m-1} < a_m$
- C. $a_1 < b_1 < a_2 < b_2 < \dots < a_{m-1} < b_{m-1} < a_m < b_m < b_{m+1} < \dots < b_{n-1} < b_n$
- D. $b_1 < b_2 < \dots < b_{m-1} < b_m < a_1 < a_2 < \dots < a_{m-1} < a_m < b_{m+1} < \dots < b_{n-1} < b_n$

50.以下关于无向连通图 G 的叙述中, 不正确的是 ()。

- A. G 中任意两个顶点之间均有边存在
- B. G 中任意两个顶点之间存在路径
- C. 从 G 中任意顶点出发可遍历图中所有顶点
- D. G 的邻接矩阵是对称矩阵

51.设某二叉树采用二叉链表表示(即结点的两个指针分别指示左、右孩子)。当该二叉树包含 k 个结点时, 其二叉链表结点中必有 () 个空的孩子指针。

- A. k-1
- B. k
- C. k+1
- D. 2k

52.假设某消息中只包含 7 个字符{a, b, c, d, e, f, g}, 这 7 个字符在消息中出现的次数为{5, 24, 8, 17, 34, 4, 13}, 利用哈夫曼树(最优二叉树)为该消息中的字符构造符合前缀编码要求的不等长编码。各字符的编码长度分别为 ()。

- A. a:4, b:2, c:3, d:3, e:2, f:4, g:3
- B. a:6, b:2, c:5, d:3, e:1, f:6, g:4
- C. a:3, b:3, c:3, d:3, e:3, f:2, g:3
- D. a:2, b:6, c:3, d:5, e:6, f:1, g:4

53.设 S 是一个长度为 n 的非空字符串, 其中的字符各不相同, 则其互异的非平凡子串(非空且不同于 S 本身)个数为 ()。

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

54.某汽车加工工厂有两条装配线 L1 和 L2, 每条装配线的工位数为 n (S_{ij} , $i=1$ 或 2, $j=1, 2, \dots, n$), 两条装配线对应的工位完成同样的加工作, 但是所需要的时间可能不同 (a_{ij} , $i=1$ 或 2, $j=1, 2, \dots, n$)。汽车底盘开始到进入两条装配线的时间 (e_1, e_2) 以及装配后到结束的时间(X_1X_2)也可能不相同。从一个工位加工后流到下一个工位需要迁移时间(t_{ij} , $i=1$ 或 2, $j=2, \dots, n$)。现在要以最快的时间完成一辆汽车的装配, 求最优的装配路线。

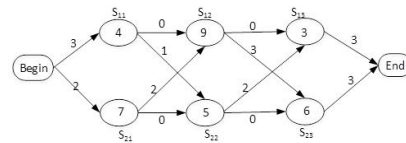
分析该问题, 发现问题具有最优子结构。以 L1 为例, 除了第一个工位之外, 经过第 j 个工位的最短时间包含了经过 L1 的第 j-1 个工位的最短时间或者经过 L2 的第 j-1 个工位的最短时间, 如式(1)。装配后到结束的最短时间包含离开 L1 的最短时间或者离开 L2 的最短时间如式(2)。

$$f_{1j} = \begin{cases} e_1 + a_{1j} & \text{若 } j=1 \\ \min(f_{1j-1} + a_{1j} + t_{1j-1}, f_{2j-1} + a_{1j} + t_{2j-1}) & \text{其他} \end{cases} \quad (1)$$

由于在求解经过 L1 和 L2 的第 j 个工位的最短时间均包含了经过 L1 的第 j-1 个工位的最短时间或者经过 L2 的第 j-1 个工位的最短时间，该问题具有重复子问题的性质，故采用迭代方法求解。

该问题采用的算法设计策略是（ ），算法的时间复杂度为（ ）

以下是一个装配调度实例，其最短的装配时间为（ ），装配路线为（ ）



- 问题 1: A.分治 B.动态规划 C.贪心 D.回溯
 问题 2: A. $\theta(\lg n)$ B. $\theta(n)$ C. $\theta(n^2)$ D. $\theta(n \lg n)$
 问题 3: A.21 B.23 C.20 D.26
 问题 4: A.S11→S12→S13 B.S11→S22→S13 C.S21→S12→S23 D.S21→S22→S23

55.在 12 个互异元素构成的有序数组 $a[1..12]$ 中进行二分查找（即折半查找，向下取整），若待查找的元素正好等于 $a[9]$ ，则在此过程中，依次与数组中的（ ）比较后，查找成功结束。

- A. $a[6]$ 、 $a[7]$ 、 $a[8]$ 、 $a[9]$ B. $a[6]$ 、 $a[9]$
 C. $a[6]$ 、 $a[7]$ 、 $a[9]$ D. $a[6]$ 、 $a[8]$ 、 $a[9]$

56.对于 n 个元素的关键字序列 $\{k_1, k_2, \dots, k_n\}$ ，当且仅当满足关系 $k_i \leq k_{2i}$ 且 $k_i \leq k_{2i+1}$ $\{i=1, 2, \dots, \lfloor n/2 \rfloor\}$ 时称其为小根堆（小顶堆）。以下序列中，（ ）不是小根堆。

- A.16, 25, 40, 55, 30, 50, 45 B.16, 40, 25, 50, 45, 30, 55
 C.16, 25, 39, 41, 45, 43, 50 D.16, 40, 25, 53, 39, 55, 45

57.某二叉树的先序遍历序列为 ABCDEF，中序遍历序列为 BADCFE，则该二叉树的高度（即层数）为（ ）。

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

58.已知栈 S 初始为空，用 I 表示入栈、 O 表示出栈，若入栈序列为 $a_1 a_2 a_3 a_4 a_5$ ，则通过栈 S 得到出栈序列 $a_2 a_4 a_5 a_3 a_1$ 的合法操作序列（ ）。

- A.IIOIOIOIOOO B.IOIOIOIOIOIO C.IOOIOIOIOIO D.IIOOIOIOIOOO

59.以下关于字符串的叙述中，正确的是（ ）。

- A.包含任意个空格字符的字符串称为空串 B.字符串不是线性数据结构
 C.字符串的长度是指串中所含字符的个数 D.字符串的长度是指串中所含非空格字符的个数

60.两个矩阵 $A_{m \times n}$ 和 $B_{n \times p}$ 相乘，用基本的方法进行，则需要的乘法次数为 $m \times n \times p$ 。多个矩阵相乘满足结合律，不同的乘法顺序所需要的乘法次数不同。考虑采用动态规划方法确定 $M_1, M_{(i+1)}, \dots, M_j$ 多个矩阵连乘的最优顺序，即所需要的乘法次数最少。最少乘法次数用 $m[i, j]$ 表示，其递归式定义为：

$$m[i, j] = \begin{cases} 0 & i \geq j \\ \min_{i \leq k < j} \{m[i, k] + m[k+1, j] + p_{i-1} p_k p_j\} & i < j \end{cases}$$

其中 i, j 和 k 为矩阵下标，矩阵序列中 M_i 的维度为 $(p_{i-1}) \times p_i$ 采用自底向上的方法实现该算法来确定 n 个矩阵相乘的顺序，其时间复杂度为（ ）。若四个矩阵 M_1, M_2, M_3, M_4 相乘的维度序列为 2、6、3、10、3，采用上述算法求解，则乘法次数为（ ）。

- 问题 1: A. $O(n^2)$ B. $O(n^2 \lg n)$ C. $O(n^3)$ D. $O(n^3 \lg n)$
 问题 2: A.156 B.144 C.180 D.360

61.下表为某文件中字符的出现频率,采用霍夫曼编码对下列字符编码,则字符序列“bee”的编码为();
编码“110001001101 ”的对应的字符序列为()。

字符	a	b	c	d	e	f
频率(%)	45	13	12	16	9	5

问题 1: A.10111011101 B.10111001100 C.001100100 D.110011011

问题 2: A.bad B.bee C.face D.bace

62.以下关于二叉排序树(或二叉查找树、二叉搜索树)的叙述中,正确的是()。

- A.对二叉排序树进行先序、中序和后序遍历,都得到结点关键字的有序序列
- B.含有 n 个结点的二叉排序树高度为 $\lceil \log_2 n \rceil + 1$
- C.从根到任意一个叶子结点的路径上,结点的关键字呈现有序排列的特点
- D.从左到右排列同层次的结点,其关键字呈现有序排列的特点

63.具有 3 个结点的二叉树有()种形态。

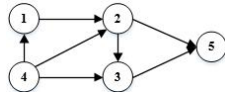
A.2 B.3 C.5 D.7

64.设有一个包含 n 个元素的有序线性表。在等概率情况下删除其中的一个元素,若采用顺序存储结构,则平均需要移动()个元素;若采用单链表存储,则平均需要移动()个元素。

问题 1: A.1 B. $(n-1)/2$ C. $\log n$ D. n

问题 2: A.0 B.1 C. $(n-1)/2$ D. $n/2$

65.拓扑序列是有向无环图中所有顶点的一个线性序列,若有向图中存在弧 $\langle v, w \rangle$ 或存在从顶点 v 到 w 的路径,则在该有向图的任一拓扑序列中, v 一定在 w 之前。下面有向图的拓扑序列是()。



A.41235 B.43125 C.42135 D.41325

66.二维数组 $a[1..N, 1..N]$ 可以按行存储或按列存储。对于数组元素 $a[i, j]$ ($1 \leq i, j \leq N$), 当()时,在按行和按列两种存储方式下,其偏移量相同。

A. $i \neq j$ B. $i = j$ C. $i > j$ D. $i < j$

67.考虑一个背包问题,共有 $n=5$ 个物品,背包容量为 $W=10$,物品的重量和价值分别为: $w=\{2, 2, 6, 5, 4\}$, $v=\{6, 3, 5, 4, 6\}$,求背包问题的最大装包价值。若此为 0-1 背包问题,分析该问题具有最优子结构,定义递归式为

$$c[i, j] = \begin{cases} 0 & \text{若 } i=0 \text{ 或 } j=0 \\ c[i-1, j] & \text{若 } w[i] > j \\ \max\{c[i-1, j], c[i-1, j-w(i)]+v(i)\} & \text{其他} \end{cases}$$

其中 $c(i, j)$ 表示 i 个物品、容量为 j 的 0-1 背包问题的最大装包价值,最终要求解 $c(n, W)$ 。采用自底向上的动态规划方法求解,得到最大装包价值为(),算法的时间复杂度为()。若此为部分背包问题,首先采用归并排序算法,根据物品的单位重量价值从大到小排序,然后依次将物品放入背包直至所有物品放入背包中或者背包再无容量,则得到的最大装包价值为(),算法的时间复杂度为()。

问题 1: A.11 B.14 C.15 D.16.67

问题 2: A. $O(nW)$ B. $O(n \lg n)$ C. $O(n^2)$ D. $O(n \lg nW)$

问题 3: A.11 B.14 C.15 D.16.67

问题 4: A. $O(nW)$ B. $O(n \lg n)$ C. $O(n^2)$ D. $O(n \lg nW)$

68.以下关于图的遍历的叙述中，正确的是（ ）。

- A. 图的遍历是从给定的源点出发对每一个顶点仅访问一次的过程
- B. 图的深度优先遍历方法不适用于无向图
- C. 使用队列对图进行广度优先遍历
- D. 图中有回路时则无法进行遍历

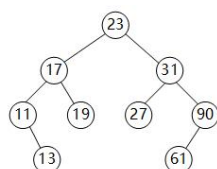
69.在 13 个元素构成的有序表 A[1..13]中进行折半查找（或称为二分查找，向下取整）。那么以下叙述中，错误的是（ ）。

- A. 无论要查找哪个元素，都是先与 A[7]进行比较
- B. 若要查找的元素等于 A[9]，则分别需与 A[7]、A[11]、A[9]进行比较
- C. 无论要查找的元素是否在 A[]中，最多与表中的 4 个元素比较即可
- D. 若待查找的元素不在 A[]中，最少需要与表中的 3 个元素进行比较

70.若一棵二叉树的高度（即层数）为 h，则该二叉树（ ）。

- A. 有 2^h 个结点
- B. 有 $2^h - 1$ 个结点
- C. 最少有 $2^h - 1$ 个结点
- D. 最多有 $2^h - 1$ 个结点

71.设有二叉排序树（或二叉查找树）如下图所示，建立该二叉树的关键码序列不可能是（ ）。



- A. 23 31 17 19 11 27 13 90 61
- B. 23 17 19 31 27 90 61 11 13
- C. 23 17 27 19 31 13 11 90 61
- D. 23 31 90 61 27 17 19 11 13

72.若元素以 a, b, c, d, e 的顺序进入一个初始为空的栈中，每个元素进栈、出栈各 1 次，要求出栈的第一个元素为 d，则合法的出栈序列共有（ ）种。

- A. 4
- B. 5
- C. 6
- D. 24

73.在某应用中，需要先排序一组大规模的记录，其关键字为整数。若这组记录的关键字基本上有序，则适宜采用（ ）排序算法。若这组记录的关键字的取值均在 0 到 9 之间（含），则适宜采用（ ）排序算法。

问题 1: A. 插入 B. 归并 C. 快速 D. 计数

问题 2: A. 插入 B. 归并 C. 快速 D. 计数

74.已知算法 A 的运行时间函数为 $T(n) = 8T(n/2) + n^2$ ，其中 n 表示问题的规模，则该算法的时间复杂度为（ ）。另已知算法 B 的运行时间函数为 $T(n) = XT(n/4) + n^2$ ，其中 n 表示问题的规模。对充分大的 n，若要算法 B 比算法 A 快，则 X 的最大值为（ ）。

问题 1: A. $O(n)$ B. $O(n \lg n)$ C. $O(n^2)$ D. $O(n^3)$

问题 2: A. 15 B. 17 C. 63 D. 65

75.设一个包含 n 个顶点、e 条弧的简单有向图采用邻接矩阵存储结构（即矩阵元素 A[i][j] 等于 1 或 0，分别表示顶点 i 与顶点 j 之间有弧或无弧），则该矩阵结构非零元素数目为（ ）。

- A. e
- B. 2e
- C. n-e
- D. n+e

76.在 55 个互异元素构成的有序表 $A[1..55]$ 中进行折半查找（或二分查找，向下取整）。若需要找的元素等于 $A[19]$ ，则在查找过程中参与比较的元素依次为（ ）、 $A[19]$ 。

- A. $A[28]$ 、 $A[30]$ 、 $A[15]$ 、 $A[20]$ B. $A[28]$ 、 $A[14]$ 、 $A[21]$ 、 $A[17]$
C. $A[28]$ 、 $A[15]$ 、 $A[22]$ 、 $A[18]$ D. $A[28]$ 、 $A[18]$ 、 $A[22]$ 、 $A[20]$

77.对于非空的二叉树，设 D 代表根结点，L 代表根结点的左子树 R 代表根结点的右子树。若对下图所示的二叉树进行遍历后的结点序列为 7 6 5 4 3 2 1，则遍历方式是（ ）。

- A. LRD B. DRL C. RLD D. RDL

78.

设某 n 阶三对角矩阵 $A_{n \times n}$ 的示意图如下图所示。若将该三对角矩阵的非零元素按行存储在一维数组 $B[k]$ ($1 \leq k \leq 3 \cdot n - 2$) 中，则 k 与 i 、 j 的对应关系是（ ）。

$$A_{n \times n} = \begin{bmatrix} a_{1,1} & a_{1,2} & & & & \\ a_{2,1} & a_{2,2} & a_{2,3} & & & \\ & a_{3,2} & a_{3,3} & a_{3,4} & & \\ & & \dots & \dots & \dots & \\ & 0 & & & & \\ & & \dots & \dots & \dots & \\ & & & a_{n,n-1} & a_{n,n} & \end{bmatrix}$$

- A. $k=2i+j-2$ B. $k=2i-j+2$ C. $k=3i+j-1$ D. $k=3i-j+2$

79.对于一个长度为 n ($n > 1$) 且元素互异的序列，令其所有元素依次通过一个初始为空的栈后，再通过一个初始为空的队列。假设队列和栈的容量都足够大，且只要栈非空就可以进行出栈操作，只要队列非空就可以进行出队操作，那么以下叙述中，正确的是（ ）。

- A. 出队序列和出栈序列一定互为逆序 B. 出队序列和出栈序列一定相同
C. 入栈序列与入队序列一定相同 D. 入栈序列与入队序列一定互为逆序

80.在 n 个数的数组中确定其第 i ($1 \leq i \leq n$) 小的数时，可以采用快速排序算法中的划分思想，对 n 个元素划分，先确定第 k 小的数，根据 i 和 k 的大小关系，进一步处理，最终得到第 i 小的数。划分过程中，最佳的基准元素选择的方法是选择待划分数组的（ ）元素。此时，算法在最坏情况下的时间复杂度为（不考虑所有元素均相等的情况）（ ）。

- 问题 1: A. 第一个 B. 最后一个 C. 中位数 D. 随机一个
问题 2: A. $O(n)$ B. $O(\lg n)$ C. $O(n \lg n)$ D. $O(n^2)$

81.优先队列通常采用（ ）数据结构实现，向优先队列中插入一个元素的时间复杂度为（ ）。

- 问题 1: A. 堆 B. 栈 C. 队列 D. 线性表
问题 2: A. $O(n)$ B. $O(1)$ C. $O(\lg n)$ D. $O(n^2)$

82.用某排序方法对一元素序列进行非递减排序时，若该方法可保证在排序前后排序码相同者的相对位置不变，则称该排序方法是稳定的。简单选择排序法排序方法是不稳定的，（ ）可以说明这个性质。

- A. 21 48 21* 63 17 B. 17 21 21* 48 63
C. 63 21 48 21* 17 D. 21* 17 48 63 21

83.对某有序顺序表进行折半查找时，（ ）不可能构成查找过程中关键字的比较序列。

- A. 45, 10, 30, 18, 25 B. 45, 30, 18, 25, 10
C. 10, 45, 18, 30, 25 D. 10, 18, 25, 30, 45

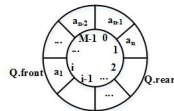
84.某二叉树的先序遍历序列为 $c a b f e d g$ ，中序遍历序列为 $a b c d e f g$ ，则该二叉树是（ ）。

- A. 完全二叉树 B. 最优二叉树 C. 平衡二叉树 D. 满二叉树

85.设栈 S 和队列 Q 的初始状态为空，元素 $a b c d e f g$ 依次进入栈 S 。要求每个元素出栈后立即进入队列 Q ，若 7 个元素出队列的顺序为 $b d f e c a g$ ，则栈 S 的容量最小应该是（ ）。

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

86.设某循环队列 Q 的定义中有 $front$ 和 $rear$ 两个域变量，其中， $front$ 指示队头元素的位置， $rear$ 指示队尾元素之后的位置，如下图所示。若该队列的容量为 M ，则其长度为（ ）。



- A. $(Q.rear - Q.front + 1)$ B. $(Q.rear - Q.front + M)$
C. $(Q.rear - Q.front + 1) \% M$ D. $(Q.rear - Q.front + M) \% M$

87.已知一个文件中出现的各字符及其对应的频率如下表所示。若采用定长编码，则该文件中字符的码长应为（ ）。若采用 Huffman 编码，则字符序列“face”的编码应为（ ）。

字符	a	b	c	d	e	f
频率 (%)	45	13	12	16	9	5

问题 1: A.2 B.3 C.4 D.5

问题 2: A.110001001101 B.001110110011 C.101000010100 D.010111101011

88.对一待排序序列分别进行直接插入排序和简单选择排序，若待排序序列中有两个元素的值相同，则（ ）保证这两个元素在排序前后的相对位置不变。

- A. 直接插入排序和简单选择排序都可以 B. 直接插入排序和简单选择排序都不能
C. 只有直接插入排序可以 D. 只有简单选择排序可以

89.快速排序算法在排序过程中，在待排序数组中确定一个元素为基准元素，根据基准元素把待排序数组划分成两个部分，前面一部分元素值小于等于基准元素，而后面一部分元素值大于基准元素。然后再分别对前后两个部分进一步进行划分。根据上述描述，快速排序算法采用了（ ）算法设计策略。已知确定基准元素操作的时间复杂度为 $\Theta(n)$ ，则快速排序算法的最好和最坏情况下的时间复杂度为（ ）。

问题 1: A. 分治 B. 动态规划 C. 贪心 D. 回溯

问题 2:

- A. $\Theta(n)$ 和 $\Theta(n \lg n)$ B. $\Theta(n)$ 和 $\Theta(n^2)$
C. $\Theta(n \lg n)$ 和 $\Theta(n \lg n)$ D. $\Theta(n \lg n)$ 和 $\Theta(n^2)$

90.在字符串的 KMP 模式匹配算法中，需先求解模式串的 $next$ 函数值，其定义如下式所示， j 表示模式串中字符的序号（从 1 开始）。若模式串 p 为“abaac”，则其 $next$ 函数值为（ ）。

$$next[j] = \begin{cases} 0 & j=1 \\ \max \{k \mid 1 < k < j, 'p_1 p_2 \dots p_{k-1}' = 'p_{j-k+1} p_{j-k+2} \dots p_{j-1}'\} & \text{其他情况} \\ 1 & \end{cases}$$

- A.01234 B.01122 C.01211 D.01111

91.某个二叉查找树（即二叉排序树）中进行查找时，效率最差的情形是该二叉查找树是（ ）。

- A. 完全二叉树 B. 平衡二叉树 C. 单枝树 D. 满二叉树

92.若一个栈初始为空，其输入序列是 1, 2, 3, ..., n-1, n, 其输出序列的第一个元素为 k ($1 \leq k \leq \lceil n/2 \rceil$)，则输出序列的最后一个元素是（ ）。

- A. 值为 n 的元素 B. 值为 1 的元素 C. 值为 n-k 的元素 D. 不确定的

93.对于线性表，相对于顺序存储，采用链表存储的缺点是（ ）。

- A. 数据元素之间的关系需要占用存储空间，导致存储密度不高
B. 表中结点必须占用地址连续的存储单元，存储密度不高
C. 插入新元素时需要遍历整个链表，运算的时间效率不高
D. 删除元素时需要遍历整个链表，运算的时间效率不高

94.Prim 算法和 Kruscal 算法都是无向连通网的最小生成树的算法，Prim 算法从一个顶点开始，每次从剩余的顶点加入一个顶点，该顶点与当前生成树中的顶点的连边权重最小，直到得到最小生成树开始，Kruscal 算法从权重最小的边开始，每次从不在当前的生成树顶点之间的边中选择权重最小的边加入，直到得到一颗最小生成树，这两个算法都采用了（ ）设计策略，且（ ）。

问题 1: A. 分治 B. 贪心 C. 动态规划 D. 回溯

问题 2:

- A. 若网较稠密，则 Prim 算法更好 B. 两个算法得到的最小生成树是一样的
C. Prim 算法比 Kruscal 算法效率更高 D. Kruscal 算法比 Prim 算法效率更高

95.在某个算法时间复杂度递归式 $T(n)=T(n-1)+n$ ，其中 n 为问题的规模，则该算法的渐进时间复杂度为（ ），若问题的规模增加了 16 倍，则运行时间增加（ ）倍。

问题 1: A. $\theta(n)$ B. $\theta(n \lg n)$ C. $\theta(n^2)$ D. $\theta(n^2 \lg n)$

问题 2: A. 16 B. 64 C. 256 D. 1024

96.实现二分查找（折半查找）时，要求查找表（ ）。

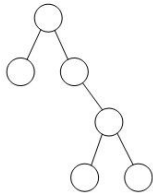
- A. 顺序存储，关键码无序排列 B. 顺序存储，关键码有序排列
C. 双向链表存储，关键码无序排列 D. 双向链表存储，关键码有序排列

97.某双端队列如下所示，要求元素进出队列必须在同一端口，即从 A 端进入的元素必须从 A 端出、从 B 端进入的元素必须从 B 端出，则对于 4 个元素的序列 e1、e2、e3、e4，若要求从前 2 个元素（e1、e2）从 A 端口按次序全部进入队列，后两个元素（e3、e4）从 B 端口按次序全部进入队列，则可能得到的出队序列是（ ）。



- A. e1、e2、e3、e4
B. e2、e3、e4、e1
C. e3、e4、e1、e2
D. e4、e3、e2、e1

98.二叉树如右图所示，若进行顺序存储（即用一维数组元素存储该二叉树中的结点且通过下标反映结点间的关系，例如，对于下标为 i 的结点，其左孩子的下标为 $2i$ 、右孩子的下标为 $2i+1$ ），则该数组的大小至少为（ ）；若采用三叉链表存储该二叉树（各个结点包括结点的数据、父结点指针、左孩子指针、右孩子指针），则该链表的所有结点中空指针的数目为（ ）。



问题 1: A.6 B.10 C.12 D.15

问题 2: A.6 B.8 C.12 D.14

99.若对线性表的最常用操作是访问任意指定序号的元素，并在表尾加入和删除元素，则适宜采用（ ）存储。

A.顺序表 B.单链表 C.双向链表 D.哈希表

100.在求解某问题时，经过分析发现该问题具有最优子结构性质，求解过程中子问题被重复求解，则采用（ ）算法设计策略；若定义问题的解空间，以深度优先的方式搜索解空间，则采用（ ）算法设计策略。

问题 1: A.分治 B.动态规划 C.贪心 D.回溯

问题 2: A.动态规划 B.贪心 C.回溯 D.分支限界

101.对 n 个基本有序的整数进行排序，若采用插入排序算法，则时间和空间复杂度分别为（ ）；若采用快速排序算法，则时间和空间复杂度分别为（ ）。

问题 1: A. $O(n^2)$ 和 $O(n)$ B. $O(n)$ 和 $O(n)$ C. $O(n^2)$ 和 $O(1)$ D. $O(n)$ 和 $O(1)$

问题 2: A. $O(n^2)$ 和 $O(n)$ B. $O(n \lg n)$ 和 $O(n)$ C. $O(n^2)$ 和 $O(1)$ D. $O(n \lg n)$ 和 $O(1)$

102.某哈希表（散列表）的长度为 n ，设散列函数为 $H(\text{Key}) = \text{Key} \bmod p$ ，采用线性探测法解决冲突。以下关于 p 值的叙述中，正确的是（ ）。

A. p 的值一般为不大于 n 且最接近 n 的质数 B. p 的值一般为大于 n 的任意整数
C. p 的值必须为小于 n 的合数 D. p 的值必须等于 n

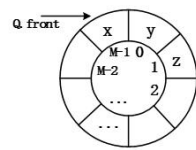
103.以下关于哈夫曼树的叙述，正确的是（ ）。

A.哈夫曼树一定是满二叉树，其每层结点数都达到最大值
B.哈夫曼树一定是平衡二叉树，其每个结点左右子树的高度差为-1、0 或 1
C.哈夫曼树中左孩子结点的权值小于父节点、右孩子节点的权值大于父节点
D.哈夫曼树中叶子节点的权值越小则距离树根越远、叶子节点的权值越大则距离树根越近

104.在一个有向图 G 的拓扑序列中，顶点 v_i 排列在 v_j 之前，说明图 G 中（ ）。

A.一定存在弧 $\langle v_i, v_j \rangle$
B.一定存在弧 $\langle v_j, v_i \rangle$
C.可能存在 v_i 到 v_j 的路径，而不可能存在 v_j 到 v_i 的路径
D.可能存在 v_j 到 v_i 的路径，而不可能存在 v_i 到 v_j 的路径

105. 设循环队列 Q 的定义中有 $front$ 和 $size$ 两个域变量，其中 $front$ 表示队头元素的指针， $size$ 表示队列的长度，如下图所示（队列长度为 3，队头元素为 x ，队尾元素为 z ）。设队列的存储空间容量为 M ，则队尾元素的指针为（ ）。



- A. $(Q.front + Q.size - 1)$ B. $(Q.front + Q.size - 1 + M) \% M$
 C. $(Q.front - Q.size)$ D. $(Q.front - Q.size + M) \% M$

106. 以下关于线性表存储结构的叙述，正确的是（ ）。

- A. 线性表采用顺序存储结构时，访问表中任意一个指定序号元素的时间复杂度为常量级
 B. 线性表采用顺序存储结构时，在表中任意位置插入新元素的运算时间复杂度为常量级
 C. 线性表采用链式存储结构时，访问表中任意一个指定序号元素的时间复杂度为常量级
 D. 线性表采用链式存储结构时，在表中任意位置插入新元素的运算时间复杂度为常量级

107. 以下关于哈希(Hash, 散列)查找叙述中，正确的是（ ）。

- A. 哈希函数应尽可能复杂些，以消除冲突
 B. 构造哈希函数时应尽量使关键字的所有组成部分都能起作用
 C. 进行哈希查找时，不再需要与查找表中的元素进行比较
 D. 在哈希表中只能添加元素不能删除元素

108. 一个高度为 h 的满二叉树的结点总数为 $2^h - 1$ ，从根结点开始，自上而下、同层次结点从左至右，对结点按照顺序依次编号，即根结点编号为 1，其左、右孩子结点编号分别为 2 和 3，再下一层从左到右的编号为 4、5、6、7，依此类推。那么，在一棵满二叉树中，对于编号为 m 和 n 的两个结点，若 $n = 2m + 1$ ，则（ ）。

- A. m 是 n 的左孩子 B. m 是 n 的右孩子 C. n 是 m 的左孩子 D. n 是 m 的右孩子

109.

给定 n 个整数构成的数组 $A = \{a_1, a_2, \dots, a_n\}$ 和整数 x ，判断 A 中是否存在两个元素 a_i 和 a_j ，使得 $a_i + a_j = x$ 。为了求解该问题，首先用归并排序算法对数组 A 进行从小到大排序；然后判断是否存在 $a_i + a_j = x$ ，具体如下列伪代码所示，则求解该问题时排序算法应用了（ ）算法设计策略，整个算法的时间复杂度为（ ）

```
i=1; j=n
while i<j
    if  $a_i + a_j = x$  return true
    else if  $a_i + a_j > x$ 
        j--;
    else
        i++;
return false;
```

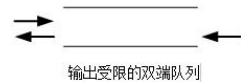
- 问题 1: A. 分治 B. 贪心 C. 动态规划 D. 回溯
 问题 2: A. $O(n)$ B. $O(n \lg n)$ C. $O(n^2)$ D. $O(n \lg^2 n)$

110.考虑下述背包问题的实例。有 5 件物品，背包容量为 100，每件物品的价值和重量如下表所示，并已经按照物品的单位重量价值从大到小排好序，根据物品单位重量价值大优先的策略装入背包中，则采用了（ ）设计策略。考虑 0/1 背包问题（每件物品或者全部放入或者全部不装入背包）和部分背包问题（物品可以部分装入背包），求解该实例，得到的最大价值分别为（ ）。

问题 1: A.分治 B.贪心 C.动态规划 D.回溯

问题 2: A.605 和 630 B.605 和 605 C.430 和 630 D.630 和 430

111.输出受限的双端队列是指元素可以从队列的两端输入，但只能从队列的一端输出，如下图所示，若有 e1, e2, e3, e4 依次进入输出受限的双端队列，则得不到输出序列（ ）。



A.e4, e3, e2, e1 B.e4, e2, e1, e3

C.e4, e3, e1, e2 D.e4, e2, e3, e1

112.设元素序列 a, b, c, d, e, f 经过初始为空的栈 S 后，得到出栈序列 cedfba，则栈 S 的最小容量为（ ）。

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

113.采用顺序表和单链表存储长度为 n 的线性序列，根据序号查找元素，其时间复杂度分别为（ ）。

A.O(1) O(1) B.O(1) O(N) C.O(N) O(1) D.O(N) O(N)

114.霍夫曼编码将频繁出现的字符采用短编码，出现频率较低的字符采用长编码。具体的操作过程为：i) 以每个字符的出现频率作为关键字构建最小优先级队列；ii) 取出关键字最小的两个结点生成子树，根节点的关键字为孩子节点关键字之和，并将根节点插入到最小优先级队列中，直至得到一颗最优编码树。

霍夫曼编码方案是基于（ ）策略的。用该方案对包含 a 到 f 六个字符的文件进行编码，文件包含 100000 个字符，每个字符的出现频率（用百分比表示）如下表所示，则与固定长度编码相比，该编码方案节省了（ ）存储空间。

字符	a	b	c	d	e	f
出现频率	18	32	4	8	12	26

问题 1: A.分治 B.贪心 C.动态规划 D.回溯

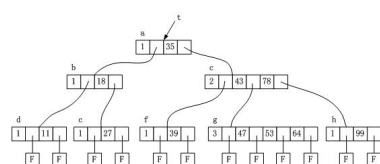
问题 2: A.21% B.27% C.18% D.36%

115.将数组{1, 1, 2, 4, 7, 5}从小到大排序，若采用（ ）排序算法，则元素之间需要进行的比较次数最少，共需要进行（ ）次元素之间的比较。

问题 1: A.直接插入 B.归并 C.堆 D.快速

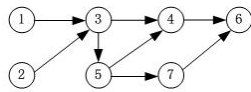
问题 2: A.5 B.6 C.7 D.8

116.下图所示为一棵 M 阶 B-树，M 最有可能的值为（ ）。



A.1 B.2 C.3 D.4

117.拓扑排序是将有向图中所有顶点排成一个线性序列的过程，并且该序列满足：若在 AOV 网中从顶点 V_i 到 V_j 有一条路径，则顶点 V_i 必然在顶点 V_j 之前。对于下面所示的有向图，（ ）是其拓扑序列。

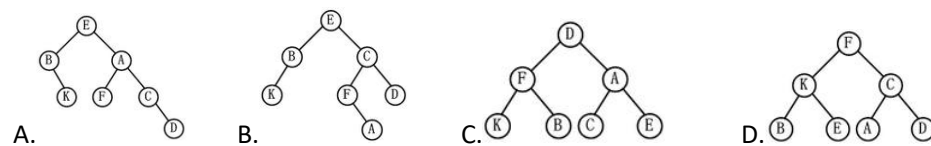


- A. 1234576 B. 1235467 C. 2135476 D. 2134567

118.在 13 个元素构成的有序表 $M[1..13]$ 中进行折半查找（向下取整），若找到的元素为 $M[4]$ ，则被比较的元素依次为（ ）。

- A. $M[7]$ 、 $M[3]$ 、 $M[5]$ 、 $M[4]$ B. $M[7]$ 、 $M[5]$ 、 $M[4]$
C. $M[7]$ 、 $M[6]$ 、 $M[4]$ D. $M[7]$ 、 $M[4]$

119. 若某二叉树的后序遍历序列为 KBFDCAE，中序遍历序列为 BKEFACD，则该二叉树为（ ）。



120.在字符串的模式匹配过程中，如果模式串的每个字符依次和主串中一个连续的字符序列相等，则称为匹配成功。如果不能在主串中找到与模式串相同的子串，则称为匹配失败。在布鲁特-福斯模式匹配算法（朴素的或基本的模式匹配）中，若主串和模式串的长度分别为 n 和 m （且 n 远大于 m ），且恰好在主串末尾的 m 个字符处匹配成功，则在上述的模式匹配过程中，字符的比较次数最多为（ ）。

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

121.现要对 n 个实数（仅包含正实数和负实数）组成的数组 A 进行重新排列，使得其中所有的负实数都位于正实数之前。求解该问题的算法的伪代码如下所示，则该算法的时间和空间复杂度分别为（ ）。

```
i=0; j=n-1;
while i<j do
    while A[i]<0 do
        i= i+1;
    while A[j]>0 do
        j =j-1;
    if i<j do
        交换 A[i]和 A[j];
```

- A. $\Theta(n)$ 和 $\Theta(n)$ B. $\Theta(1)$ 和 $\Theta(n)$ C. $\Theta(n)$ 和 $\Theta(1)$ D. $\Theta(1)$ 和 $\Theta(1)$

122.某货车运输公司有一个中央仓库和 n 个运输目的地，每天要从中央仓库将货物运输到所有运输目的地，到达每个运输目的地一次且仅一次，最后回到中央仓库。在两个地点 i 和 j 之间运输货物存在费用 C_{ij} 。为求解旅行费用总和最小的运输路径，设计如下算法：首先选择离中央仓库最近的运输目的地 1，然后选择离运输目的地 1 最近的运输目的地 2，...，每次在需访问的运输目的地中选择离当前运输目的地最近的运输目的地，最后回到中央仓库。刚该算法采用了（ ）算法设计策略，其时间复杂度为（ ）。

问题 1: A. 分治 B. 动态规划 C. 贪心 D. 回溯

问题 2: A. $\Theta(n^2)$ B. $\Theta(n)$ C. $\Theta(n \lg n)$ D. $\Theta(1)$

123.以下关于渐近符号的表示中，不正确的是（ ）。

- A. $n^2 = \Theta(n^2)$ B. $n^2 = O(n^2)$ C. $n^2 = O(n)$ D. $n^2 = O(n^3)$

124.递增序列 A (a1, a2, ..., an) 和 B (b1, b2, ..., bn) 的元素互不相同，若需将它们合并为一个长度为 2n 的递增序列，则当最终的排列结果为（ ）时，归并过程中元素的比较次数最多。

- A. a1, a2, ..., an, b1, b2, ..., bn
B. b1, b2, ..., bn, a1, a2, ..., an
C. a1, b1, a2, b2, ..., ai, bi, ..., an, bn
D. a1, a2, ..., ai/2, b1, b2, ..., bi/2, ai/2+1, ai/2+2, ..., an, bi/2+1, ..., bn

125.从存储空间的利用率角度来看，以下关于数据结构中图的存储的叙述，正确的是（ ）。

- A. 有向图适合采用邻接矩阵存储，无向图适合采用邻接表存储
B. 无向图适合采用邻接矩阵存储，有向图适合采用邻接表存储
C. 完全图适合采用邻接矩阵存储
D. 完全图适合采用邻接表存储

126.若 n_2 、 n_1 、 n_0 分别表示一个二叉树中度为 2、度为 1 和叶子结点的数目（结点的度定义为结点的子树数目），则对于任何一个非空的二叉树，（ ）。

- A. n_2 一定大于 n_1 B. n_1 一定大于 n_0 C. n_2 一定大于 n_0 D. n_0 一定大于 n_2

127.在字符串的 KMP 模式匹配算法中，需要求解模式串 p 的 next 函数值，其定义如下所示。若模式串 p 为“aaabaaa”，则其 next 函数值为（ ）。

$$next[j] = \begin{cases} 0 & j=1 \\ \max\{k | 1 \leq k < j, p_1 p_2 \dots p_{k-1} = p_{j-k+1} p_{j-k+2} \dots p_{j-1}\} & \text{其他情况} \\ 1 & \end{cases}$$

- A. 0123123 B. 0123210 C. 0123432 D. 0123456

128.对于一个长度大于 1 且不存在重复元素的序列，令其所有元素依次通过一个初始为空的队列后，再通过一个初始为空的栈。设队列和栈的容量都足够大，一个序列通过队列（栈）的含义是序列的每个元素都入队列（栈）且出队列（栈）一次且仅一次。对于该序列在上述队列和栈上的操作，正确的叙述是（ ）。

- A. 出队序列和出栈序列一定相同
B. 出队序列和出栈序列一定互为逆序
C. 入队序列与出队序列一定相同，入栈序列与出栈序列不一定相同
D. 入栈序列与出栈序列一定互为逆序，入队序列与出队序列不一定互为逆序

129.对于二维数组 a[1..N, 1..N] 中的一个元素 a[i, j] (1 ≤ i, j ≤ N)，存储在 a[i, j] 之前的元素个数（ ）。

- A. 与按行存储或按列存储方式无关 B. 在 i=j 时与按行存储或按列存储方式无关
C. 在按行存储方式下比按列存储方式下要多 D. 在按行存储方式下比按列存储方式下要少

130.设算法 A 的时间复杂度可用递归式 $T(n) = \begin{cases} \Theta(1) & n=1 \\ 7T(n/2) + n^2 & n>1 \end{cases}$ 表示，算法 B 的时间复杂度可用递归式 $T(n) = \begin{cases} \Theta(1) & n=1 \\ 3T(n/3) + n^2 & n>1 \end{cases}$

表示，若要使得算法 B 渐进地快于算法 A，则 a 的最大整数为（ ）。

- A. 48 B. 49 C. 13 D. 14

131.对 n 个元素值分别为 -1、0 或 1 的整型数组 A 进行升序排序的算法描述如下：统计 A 中 -1、0 和 1 的个数，设分别为 n_1 、 n_2 和 n_3 ，然后将 A 中的前 n_1 个元素赋值为 -1，第 n_1+1 到 n_1+n_2 个元素赋值为 0，最后 n_3 个元素赋值为 1。该算法的时间复杂度和空间复杂度分别为（ ）。

- A. $\Theta(n)$ 和 $\Theta(1)$ B. $\Theta(n)$ 和 $\Theta(n)$ C. $\Theta(n^2)$ 和 $\Theta(1)$ D. $\Theta(n^2)$ 和 $\Theta(n)$

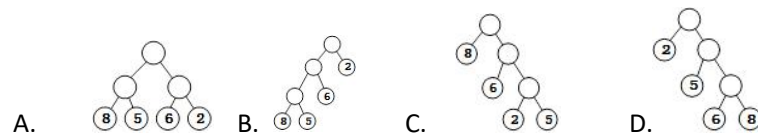
132.在有 n 个无序无重复元素值的数组中查找第 i 小的数的算法描述如下：任意取一个元素 r ，用划分操作确定其在数组中的位置，假设元素 r 为第 k 小的数。若 i 等于 k ，则返回该元素值；若 i 小于 k ，则在划分的前半部分递归进行划分操作找第 i 小的数；否则在划分的后半部分递归进行划分操作找第 $k-i$ 小的数。该算法是一种基于（ ）策略的算法。

- A. 分治 B. 动态规划 C. 贪心 D. 回溯

133.迪杰斯特拉(Dijkstra)算法用于求解图上的单源点最短路径。该算法按路径长度递增次序产生最短路径，本质上说，该算法是一种基于（ ）策略的算法。

- A. 分治 B. 动态规划 C. 贪心 D. 回溯

134.（ ）一是由权值集合 {8, 5, 6, 2} 构造的哈夫曼树（最优二叉树）。



135.一棵满二叉树，其每一层结点个数都达到最大值，对其中的结点从 1 开始顺序编号，即根结点编号为 1，其左、右孩子结点编号分别为 2 和 3，再下一层从左到右的编号为 4、5、6、7，依此类推，每一层都从左到右依次编号，直到最后的叶子结点层为止，则用（ ）可判定编号为 m 和 n 的两个结点是否在同一层。

- A. $\log_2 m = \log_2 n$ B. $\lfloor \log_2 m \rfloor = \lfloor \log_2 n \rfloor$ C. $\lfloor \log_2 m \rfloor + 1 = \lfloor \log_2 n \rfloor$ D. $\lfloor \log_2 m \rfloor = \lfloor \log_2 n \rfloor + 1$

136.无向图中一个顶点的度是指图中与该顶点相邻接的顶点数。若无向图 G 中的顶点数为 n ，边数为 e ，则所有顶点的度数之和为（ ）。

- A. $n * e$ B. $n e$ C. $2n$ D. $2e$

137.对于线性表（由 n 个同类元素构成的线性序列），采用单向循环链表存储的特点之一是（ ）。

- A. 从表中任意结点出发都能遍历整个链表
B. 对表中的任意结点可以进行随机访问
C. 对于表中的任意一个结点，访问其直接前驱和直接后继结点所用时间相同
D. 第一个结点必须是头结点

138.在 KMP 模式匹配算法中，需要求解模式串 p 的 next 函数值，其定义如下（其中， j 为模式串中字符的序号）。对于模式串“abaabaca”，其 next 函数值序列为（ ）。

$$\text{next}[j] = \begin{cases} 0 & j=1 \\ \max \{k \mid 1 < k < j, 'p_1 p_2 \dots p_{k-1}' = 'p_{j-k+1} p_{j-k+2} \dots p_{j-1}'\} & \text{其他情况} \\ 1 & \end{cases}$$

- A. 01111111 B. 01122341 C. 01234567 D. 01122334

139.若二维数组 $\text{arr}[1..M, 1..N]$ 的首地址为 base ，数组元素按列存储且每个元素占用 K 个存储单元，则元素 $\text{arr}[i, j]$ 在该数组空间的地址为（ ）。

- A. $\text{base} + ((i-1)*M+j-1)*K$ B. $\text{base} + ((i-1)*N+j-1)*K$
C. $\text{base} + ((j-1)*M+i-1)*K$ D. $\text{base} + ((j-1)*N+i-1)*K$

140.用插入排序和归并排序算法对数组 $\langle 3, 1, 4, 1, 5, 9, 6, 5 \rangle$ 进行从小到大排序，则分别需要进行（ ）次数组元素之间的比较。

- A. 12, 14 B. 10, 14 C. 12, 16 D. 10, 16

141.某算法的时间复杂度可用递归式 $T(n) = \begin{cases} \Theta(1) & , n=1 \\ 6T(n/5) + n & , n>1 \end{cases}$ 表示，若用 Θ 表示，则正确的是（ ）。

- A. $\Theta(n^{\log_5 6})$ B. $\Theta(n^2)$ C. $\Theta(n)$ D. $\Theta(n^{\log_5 6})$

142.分治算法设计技术（ ）。

- A. 一般由三个步骤组成：问题划分、递归求解、合并解 B. 一定是用递归技术来实现
C. 将问题划分为 k 个规模相等的子问题 D. 划分代价很小而合并代价很大

143.要在 $8*8$ 的棋盘上摆放 8 个“皇后”，要求“皇后”之间不能发生冲突，即任何两个“皇后”不能在同一行、同一列和相同的对角线上，则一般采用（ ）来实现。

- A. 分治法 B. 动态规划法 C. 贪心法 D. 回溯法

144.对于关键字序列 $\langle 26, 25, 72, 38, 8, 18, 59 \rangle$ ，采用散列函数 $H(\text{Key}) = \text{Key} \bmod 13$ 构造散列表（哈希表）。若采用线性探测的开放定址法解决冲突（顺序地探查可用存储单元），则关键字 59 所在散列表中的地址为（ ）。

- A. 6 B. 7 C. 8 D. 9

145.设一个包含 N 个顶点、 E 条边的简单无向图采用邻接矩阵存储结构（矩阵元素 $A[i][j]$ 等于 1/0 分别表示顶点 i 与顶点 j 之间有 / 无边），则该矩阵中的非零元素数目为（ ）。

- A. N B. E C. $2E$ D. $N+E$

146.在（ ）中，任意一个结点的左、右子树的高度之差的绝对值不超过 1。

- A. 完全二叉树 B. 二叉排序树 C. 线索二叉树 D. 最优二叉树

147.对 n 个元素的有序表 $A[1..n]$ 进行顺序查找，其成功查找的平均查找长度（即在查找表中找到指定关键码的元素时，所进行比较的表中元素个数的期望值）为（ ）。

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

148.设下三角矩阵（上三角部分的元素值都为 0） $A[0..n, 0..n]$ 如下所示，将该三角矩阵的所有非零元素（即行下标不小于列下标的元素）按行优先压缩存储在容量足够大的数组 $M[]$ 中（下标从 1 开始），则元素 $A[i, j]$ ($0 \leq i \leq n, j \leq i$) 存储在数组 M 的（ ）中。

$$\begin{bmatrix} A_{00} & & & & & \\ A_{01} & A_{11} & & & & \\ \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & \cdot & & \\ A_{0n} & A_{1n} & A_{2n} & \cdots & \cdots & A_{nn} \\ A_{0n} & A_{1n} & A_{2n} & A_{3n} & \cdots & A_{nn} \end{bmatrix}$$

- A. $M[\frac{i(i+1)}{2} + j + 1]$ B. $M[\frac{i(i+1)}{2} + j]$ C. $M[\frac{i(i-1)}{2} + j]$ D. $M[\frac{i(i-1)}{2} + j + 1]$

149. () 不能保证求得 0-1 背包问题的最优解。

- A. 分支限界法 B. 贪心算法 C. 回溯法 D. 动态规划策略

150. 用动态规划策略求解矩阵连乘问题 $M_1 * M_2 * M_3 * M_4$, 其中 $M_1(20*5)$ 、 $M_2(5*35)$ 、 $M_3(35*4)$ 和 $M_4(4*25)$, 则最优的计算次序为 ()。

- A. $((M_1 * M_2) * M_3) * M_4$ B. $(M_1 * M_2) * (M_3 * M_4)$ C. $(M_1 * (M_2 * M_3)) * M_4$ D. $M_1 * (M_2 * (M_3 * M_4))$

151. 某算法的时间复杂度可用递归式 $T(n) = \begin{cases} O(1) & , n=1 \\ 2T(n/2) + n \lg n & , n>1 \end{cases}$ 表示, 若用 Θ 表示该算法的渐进时间复杂度的紧致界, 则正确的是 ()。

- A. $\Theta(n \lg^2 n)$ B. $\Theta(n \lg n)$ C. $\Theta(n^2)$ D. $\Theta(n^3)$

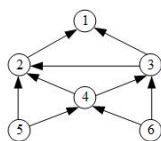
152. 已知一棵度为 3 的树 (一个结点的度是指其子树的数目, 树的度是指该树中所有结点的度的最大值) 中有 5 个度为 1 的结点, 4 个度为 2 的结点, 2 个度为 3 的结点, 那么, 该树中的叶子结点数目为 ()。

- A. 10 B. 9 C. 8 D. 7

153. 某一维数组中依次存放了数据元素 15, 23, 38, 47, 55, 62, 88, 95, 102, 123, 采用折半 (二分) 法查找元素 95 时, 依次与 () 进行了比较。

- A. 62, 88, 95 B. 62, 95 C. 55, 88, 95 D. 55, 95

154. () 是右图的合法拓扑序列。

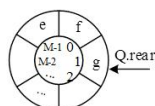


- A. 6 5 4 3 2 1 B. 1 2 3 4 5 6 C. 5 6 3 4 2 1 D. 5 6 4 2 1 3

155. 下面关于哈夫曼树的叙述中, 正确的是 ()。

- A. 哈夫曼树一定是完全二叉树
B. 哈夫曼树一定是平衡二叉树
C. 哈夫曼树中权值最小的两个结点互为兄弟结点
D. 哈夫曼树中左孩子结点小于父结点、右孩子结点大于父结点

156. 设循环队列 Q 的定义中有 $rear$ 和 len 两个域变量, 其中 $rear$ 表示队尾元素的指针, len 表示队列的长度, 如下图所示 (队列长度为 3, 队头元素为 e)。设队列的存储空间容量为 M , 则队头元素的指针为 ()。



- A. $(Q.rear + Q.len - 1)$
B. $(Q.rear + Q.len - 1 + M) \% M$
C. $(Q.rear - Q.len + 1)$
D. $(Q.rear - Q.len + 1 + M) \% M$

157.若对一个链表最常用的操作是在末尾插入结点和删除尾结点，则采用仅设尾指针的单向循环链表（不含头结点）时，（ ）。

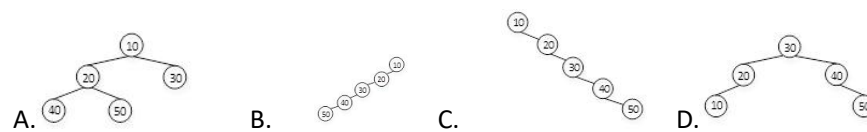
- A.插入和删除操作的时间复杂度都为 $O(1)$
- B.插入和删除操作的时间复杂度都为 $O(n)$
- C.插入操作的时间复杂度为 $O(1)$ ，删除操作的时间复杂度为 $O(n)$
- D.插入操作的时间复杂度为 $O(n)$ ，删除操作的时间复杂度为 $O(1)$

158.若某算法在问题规模为 n 时，其基本操作的重复次数可由下式表示，则该算法的时间复杂度为（ ）。

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

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

159.用关键字序列 10、20、30、40、50 构造的二叉排序树（二叉查找树）为（ ）。



160.对于哈希表，如果将装填因子 α 定义为表中装入的记录数与表的长度之比，那么向表中加入新记录时，（ ）。

- A. α 的值随冲突次数的增加而递减
- B. α 越大发生冲突的可能性就越大
- C. α 等于 1 时不会再发生冲突
- D. α 低于 0.5 时不会发生冲突

161.对以下四个序列用直接插入排序方法由小到大进行排序时，元素比较次数最少的是（ ）。

- A. 89, 27, 35, 78, 41, 15
- B. 27, 35, 41, 16, 89, 70
- C. 15, 27, 46, 40, 64, 85
- D. 90, 80, 45, 38, 30, 25

162.栈是一种按“后进先出”原则进行插入和删除操作的数据结构，因此，（ ）必须用栈。

- A. 实现函数或过程的递归调用及返回处理时
- B. 将一个元素序列进行逆置
- C. 链表结点的申请和释放
- D. 可执行程序装入和卸载

163.若用 n 个权值构造一棵最优二叉树（哈夫曼树），则该二叉树的结点总数为（ ）。

- A. $2n$
- B. $2n-1$
- C. $2n+1$
- D. $2n+2$

164.设有如下所示的下三角矩阵 $A[0..8, 0..8]$ ，将该三角矩阵的非零元素（即行下标不小于列下标的所有元素）按行优先压缩存储在数组 $M[1..m]$ 中，则元素 $A[i, j]$ ($0 \leq i \leq 8, j \leq i$) 存储在数组 M 的（ ）中。

$$\begin{bmatrix} A_{0,0} & & & & & & & & \\ A_{1,0} & A_{1,1} & & & & & & & \\ \cdot & & \cdot & & & & & & \\ \cdot & & & \cdot & & & & & \\ \cdot & & & & \cdot & & & & \\ A_{7,0} & A_{7,1} & A_{7,2} & \dots & & & A_{7,7} & & \\ A_{8,0} & A_{8,1} & A_{8,2} & A_{8,3} & \dots & & & & A_{8,8} \end{bmatrix}$$

- A. $M[\frac{i(i+1)}{2} + j + 1]$
- B. $M[\frac{i(i+1)}{2} + j]$
- C. $M[\frac{i(i-1)}{2} + j]$
- D. $M[\frac{i(i-1)}{2} + j + 1]$

165.对 n 个元素的有序表 $A[1..n]$ 进行二分(折半)查找(除 2 取商时向下取整), 查找元素 $A[i]$ ($1 \leq i \leq n$) 时, 最多与 A 中的 () 个元素进行比较。

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

166.对于具有 n 个元素的一个数据序列, 若只需得到其中第 k 个元素之前的部分排序, 最好采用 (), 使用分治(Divide and Conquer)策略的是 () 算法。

- 问题 1: A. 希尔排序 B. 直接插入排序 C. 快速排序 D. 堆排序
问题 2: A. 冒泡排序 B. 插入排序 C. 快速排序 D. 堆排序

167. () 算法策略与递归技术的联系最弱。

- A. 动态规划 B. 贪心 C. 回溯 D. 分治

168.求单源点最短路径的迪杰斯特拉(Dijkstra)算法是按 () 的顺序求源点到各顶点的最短路径的。

- A. 路径长度递减 B. 路径长度递增 C. 顶点编号递减 D. 顶点编号递增

169.对于 n 个元素的关键字序列 $\{k_1, k_2, \dots, k_n\}$, 当且仅当满足关系 $k_i \leq k_{2i}$ 且 $k_i \leq k_{2i+1}$ ($2i \leq n, 2i+1 \leq n$) 称其为小根堆, 反之则为大根堆。以下序列中, () 不符合堆的定义。

- A. (4, 10, 15, 72, 39, 23, 18) B. (58, 27, 36, 12, 8, 23, 9)
C. (4, 10, 18, 72, 39, 23, 15) D. (58, 36, 27, 12, 8, 23, 9)

170. () 在其最好情况下的算法时间复杂度为 $O(n)$ 。

- A. 插入排序 B. 归并排序 C. 快速排序 D. 堆排序

171.设某算法的计算时间可用递推关系式 $T(n) = 2T(n/2) + n$ 表示, 则该算法的时间复杂度为 ()。

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

172.在 () 存储结构中, 数据结构中元素的存储地址与其关键字之间存在某种映射关系。

- A. 顺序(Sequence) B. 链表(Link) C. 索引(Index) D. 散列(Hash)

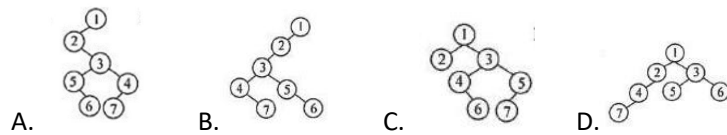
173.在平衡二叉树中, ()。

- A. 任意结点的左、右子树结点数目相同
B. 任意结点的左、右子树高度相同
C. 任意结点的左右子树高度之差的绝对值不大于 1
D. 不存在度为 1 的结点

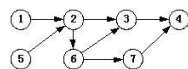
174.给定一个有 n 个元素的有序线性表。若采用顺序存储结构, 则在等概率前提下, 删除其中的一个元素平均需要移动 () 个元素。

- A. $\frac{(n+1)}{2}$ B. $\frac{n}{2}$
C. $\frac{(n-1)}{2}$ D. 1

175.为便于存储和处理一般树结构形式的信息，常采用孩子-兄弟表示法将其转换成二叉树（左子关系表示父子、右子关系表示兄弟），与下图所示的树对应的二叉树是（ ）。

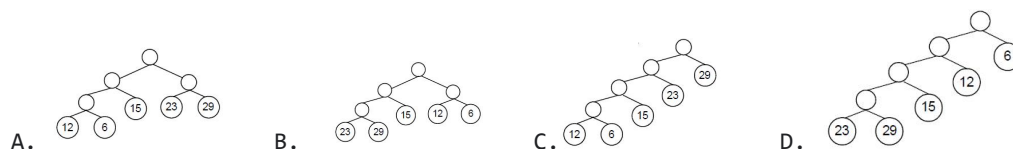


176.拓扑序列是无环有向图中所有顶点的一个线性序列，图中任意路径中的各个顶点在该图的拓扑序列中保持先后关系，（ ）为下图所示有向图的一个拓扑序列。



- A. 1 2 3 4 5 6 7 B. 1 5 2 6 3 7 4 C. 5 1 2 6 3 4 7 D. 5 1 2 3 7 6 4

177.由权值为 29、12、15、6、23 的五个叶子结点构造的哈夫曼树为（ ），其带权路径长度为（ ）。
问题 1:



问题 2: A.85 B.188 C.192 D.222

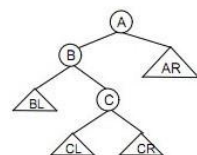
178.对 n 个元素的数组进行（ ），其平均时间复杂度和最坏情况下的时间复杂度都是 $O(n\log n)$ 。

- A. 希尔排序 B. 快速排序 C. 堆排序 D. 选择排序

179.设商店有 10 元、5 元、2 元和 1 元的零币，每种零币的数量充足。售货员给顾客找零钱时，零币的数量越少越好。例如给顾客找零 29 元：先选 2 张 10 元币，然后选择 1 张 5 元币，再选择两张 2 元币。以上的找零钱方法采用了（ ）策略。

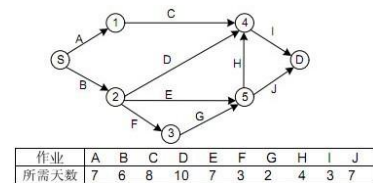
- A. 分治 B. 贪心 C. 动态规划 D. 回溯

180.下图所示平衡二叉树（树中任一结点的左右子树高度之差不超过 1）中，结点 A 的右子树 AR 高度为 h ，结点 B 的左子树 BL 高度为 h ，结点 C 的左子树 CL、右子树 CR 高度都为 $h-1$ 。若在 CR 中插入一个结点并使 CR 的高度增加 1，则该二叉树（ ）。



- A. 以 B 为根的子二叉树变为不平衡
B. 以 C 为根的子二叉树变为不平衡
C. 以 A 为根的子二叉树变为不平衡
D. 仍然是平衡二叉树

181.某工程计划如下图所示，各个作业所需的天数如下表所示，设该工程从第 0 天开工，则该工程的最短工期是（ ）天，作业 J 最迟应在第（ ）天开工。



问题 1: A.17 B.18 C.19 D.20

问题 2: A.11 B.13 C.14 D.16

182.已知某二叉树的中序序列为 CBDAEFI、先序序列为 ABCDEFI，则该二叉树的高度为（ ）。

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

183.某虚拟存储系统采用最近最少使用（LRU）页面淘汰算法，假定系统为每个作业分配 3 个页面的主存空间，其中一个页面用来存放程序。现有某作业的部分语句如下：

```
Var A: Array[1..150, 1..100] OF integer;
i, j: integer;
FOR i:=1 to 150 DO
  FOR j:=1 to 100 DO
```

A[i, j]:=0;

设每个页面可存放 150 个整数变量，变量 i、j 放在程序页中。初始时，程序及变量 i、j 已在内存，其余两页为空，矩阵 A 按行序存放。在上述程序片段执行过程中，共产生（ ）次缺页中断。最后留在内存中的是矩阵 A 的最后（ ）。

问题 1: A.50 B.100 C.150 D.300

问题 2: A.2 行 B.2 列 C.3 行 D.3 列

184.设某算法的计算时间表示为递推关系式 $T(n) = T(n-1) + n$ ($n > 0$) 及 $T(0) = 1$ ，则该算法的时间复杂度为（ ）。

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

185.给定一组长度为 n 的无序序列，将其存储在一维数组 a[0..n-1] 中。现采用如下方法找出其中的最大元素和最小元素：比较 a[0] 和 a[n-1]，若 a[0] 较大，则将二者的值进行交换；再比较 a[1] 和 a[n-2]，若 a[1] 较大，则交换二者的值；然后依次比较 a[2] 和 a[n-3]、a[3] 和 a[n-4]、...，使得每一对元素中的较小者被交换到低下标端。重复上述方法，在数组的前 n/2 个元素中查找最小元素，在后 n/2 个元素查找最大元素，从而得到整个序列的最小元素和最大元素。上述方法采用的算法设计策略是（ ）。

A.动态规划法 B.贪心法 C.分治法 D.回溯法

186.具有 n 个顶点、e 条边的图采用邻接表存储结构，进行深度优先遍历和广度优先遍历运算的时间复杂度均为（ ）。

A. $O(n^2)$ B. $O(e^2)$ C. $O(n * e)$ D. $O(n + e)$

187.某一维数组中依次存放了数据元素 12, 23, 30, 38, 41, 52, 54, 76, 85，在用折半（二分）查找方法（向上取整）查找元素 54 时，所经历“比较”运算的数据元素依次为（ ）。

A.41, 52, 54 B.41, 76, 54 C.41, 76, 52, 54 D.41, 30, 76, 54

188.广义表中的元素可以是原子，也可以是表，因此广义表的适用存储结构是（）。

- A.链表 B.静态数组 C.动态数组 D.散列表

189.将一个无序序列中的元素依次插入到一棵（），并进行中序遍历，可得到一个有序序列。

- A.完全二叉树 B.最小生成树 C.二叉排序树 D.最优二叉树

190.（）的邻接矩阵是一个对称矩阵。

- A.无向图 B.AOV 网 C.AOE 网 D.有向图

191.一个具有 m 个结点的二叉树，其二叉链表结点（左、右孩子指针分别用 **left** 和 **right** 表示）中的空指针总数必定为（）个。为形成中序（先序、后序）线索二叉树，现对该二叉链表所有结点进行如下操作：若结点 p 的左孩子指针为空，则将该左指针改为指向 p 在中序（先序、后序）遍历序列的前驱结点；若 p 的右孩子指针为空，则将该右指针改为指向 p 在中序（先序、后序）遍历序列的后继结点。假设指针 s 指向中序（先序、后序）线索二叉树中的某结点，则（）。

- 问题 1: A. $m+2$ B. $m+1$ C. m D. $m-1$

问题 2:

- A. $s \rightarrow \text{right}$ 指向的结点一定是 s 所指结点的直接后继结点
B. $s \rightarrow \text{left}$ 指向的结点一定是 s 所指结点的直接前驱结点
C.从 s 所指结点出发的 **right** 链可能构成环
D. s 所指结点的 **left** 和 **right** 指针一定指向不同的结点

192.若总是以待排序列的第一个元素作为基准元素进行快速排序，那么最好情况下的时间复杂度为（）。

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

193.斐波那契（Fibonacci）数列可以递归地定义为：

$$F(n) = \begin{cases} 1 & n = 0 \\ 1 & n = 1 \\ F(n-1) + F(n-2) & n > 1 \end{cases}$$

用递归算法求解 $F(5)$ 时需要执行（）次“+”运算，该方法采用的算法策略是（）。

- 问题 1: A.5 B.6 C.7 D.8

- 问题 2: A.动态规划 B.分治 C.回溯 D.分支限界

194.一个算法是对某类给定问题求解过程的精确描述，算法中描述的操作都可以通过将已经实现的基本操作执行有限次来实现，这句话说明算法具有（）特性。

- A.有穷性 B.可行性 C.确定性 D.健壮性

195.设一个包含 N 个顶点、 E 条边的简单有向图采用邻接矩阵存储结构（矩阵元素 $A[i][j]$ 等于 1/0 分别表示顶点 i 与顶点 j 之间有/无弧），则该矩阵的元素数目为（），其中非零元素数目为（）。

问题 1:

- A. E^2 B. N^2

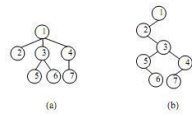
- C. $N^2 - E^2$ D. $N^2 + E^2$

问题 2:

- A. N B. $N+E$

- C. E D. $N-E$

196.若将某有序树 T 转换为二叉树 T_1 , 则 T 中结点的后(根)序序列就是 T_1 中结点的()遍历序列。例如, 下图(a)所示的有序树转化为二叉树后如图(b)所示。



- A. 先序 B. 中序 C. 后序 D. 层序

197.已知一个线性表(16, 25, 35, 43, 51, 62, 87, 93), 采用散列函数 $H(\text{Key}) = \text{Key} \bmod 7$ 将元素散列到表长为9的散列表中。若采用线性探测的开放定址法解决冲突(顺序地探查可用存储单元), 则构造的哈希表为(), 在该散列表上进行等概率成功查找的平均查找长度为() (为确定记录在查找表中的位置, 需和给定关键字值进行比较的次数的期望值称为查找算法在查找成功时的平均查找长度)。

问题 1:

A.

0	1	2	3	4	5	6	7	8
35	43	16	51	25		62	87	93

B.

0	1	2	3	4	5	6	7	8
35	43	16	93	25	51	62	87	

C.

0	1	2	3	4	5	6	7	8
35	43	16	51	25	87	62	93	

D.

0	1	2	3	4	5	6	7	8
35	43	16	51	25	87	62		93

问题 2: A. $(5*1+2+3+6) / 8$ B. $(5*1+2+3+6) / 9$ C. $(8*1) / 8$ D. $(8*1) / 9$

198.若有数组声明 $a[0..3, 0..2, 1..4]$, 设编译时为 a 分配的存储空间首地址为 base_a , 且每个数组元素占据一个存储单元。当元素以行为序存放(即按 $a[0, 0, 1]$, $a[0, 0, 2]$, $a[0, 0, 3]$, $a[0, 0, 4]$, $a[0, 1, 1]$, $a[0, 1, 2]$, ..., $a[3, 2, 4]$ 顺序存储), 则数组元素 $a[2, 2, 2]$ 在其存储空间中相对 base_a 的偏移量是()。

- A.8 B.12 C.33 D.48

199.归并排序采用的算法设计方法属于()。

- A. 归纳法 B. 分治法 C. 贪心法 D. 回溯方法

200.以下的算法设计方法中, () 以获取问题最优解为目标。

- A. 回溯方法 B. 分治法 C. 动态规划 D. 递推

201.现有 16 枚外形相同的硬币, 其中有一枚比真币的重量轻的假币, 若采用分治法找出这枚假币, 至少比较() 次才能够找出该假币。

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

202.设 L 为广义表, 将 $\text{head}(L)$ 定义为取非空广义表的第一个元素, $\text{tail}(L)$ 定义为取非空广义表除第一个元素外剩余元素构成的广义表。若广义表 $L = ((x, y, z), a, (u, t, w))$, 则从 L 中取出原子项 y 的运算是()。

- A. $\text{head}(\text{tail}(\text{tail}(L)))$ B. $\text{tail}(\text{head}(\text{head}(L)))$
C. $\text{head}(\text{tail}(\text{head}(L)))$ D. $\text{tail}(\text{tail}(\text{head}(L)))$

203.下面关于二叉树的叙述，正确的是（ ）。

- A. 完全二叉树的高度 h 与其结点数 n 之间存在确定的关系
- B. 在二叉树的顺序存储和链式存储结构中，完全二叉树更适合采用链式存储结构
- C. 完全二叉树中一定不存在度为 1 的结点
- D. 完全二叉树中必定有偶数个叶子结点

204.下面关于栈和队列的叙述，错误的是（ ）。

- A. 栈和队列都是操作受限的线性表
- B. 队列采用单循环链表存储时，只需设置队尾指针就可使入队和出队操作的时间复杂度都为 $O(1)$
- C. 若队列的数据规模 n 可以确定，则采用顺序存储结构比链式存储结构效率更高
- D. 利用两个栈可以模拟一个队列的操作，反之亦可

205.下面关于二叉排序树的叙述，错误的是（ ）。

- A. 对二叉排序树进行中序遍历，必定得到结点关键字的有序序列
- B. 依据关键字无序的序列建立二叉排序树，也可能构造出单支树
- C. 若构造二叉排序树时进行平衡化处理，则根结点的左子树结点数与右子树结点数的差值一定不超过 1
- D. 若构造二叉排序树时进行平衡化处理，则根结点的左子树高度与右子树高度的差值一定不超过 1

206.下面关于图（网）的叙述，正确的是（ ）。

- A. 连通无向网的最小生成树中，顶点数恰好比边数多 1
- B. 若有向图是强连通的，则其边数至少是顶点数的 2 倍
- C. 可以采用 AOV 网估算工程的工期
- D. 关键路径是 AOE 网中源点至汇点的最短路径

207.下面关于查找运算及查找表的叙述，错误的是（ ）。

- A. 哈希表可以动态创建
- B. 二叉排序树属于动态查找表
- C. 二分查找要求查找表采用顺序存储结构或循环链表结构
- D. 顺序查找方法既适用于顺序存储结构，也适用于链表结构

208.以下关于快速排序算法的描述中，错误的是（ ）。在快速排序过程中，需要设立基准元素并划分序列来进行排序。若序列由元素 {12, 25, 30, 45, 52, 67, 85} 构成，则初始排列为（ ）时，排序效率最高（令序列的第一个元素为基准元素）。

问题 1:

- A. 快速排序算法是不稳定的排序算法
- B. 快速排序算法在最坏情况下的时间复杂度为 $O(n \lg n)$
- C. 快速排序算法是一种分治算法
- D. 当输入数据基本有序时，快速排序算法具有最坏情况下的时间复杂度

问题 2:

- A. 45, 12, 30, 25, 67, 52, 85
- B. 85, 67, 52, 45, 30, 25, 12
- C. 12, 25, 30, 45, 52, 67, 85
- D. 45, 12, 25, 30, 85, 67, 52

209.某算法的时间复杂度表达式为 $T(n)=an^2+bn\lg n+cn+d$ ，其中， n 为问题的规模， a 、 b 、 c 和 d 为常数，用 O 表示其渐近时间复杂度为（ ）。

- A. $O(n^2)$ B. $O(n)$ C. $O(n\lg n)$ D. $O(1)$

210.字符串采用链表存储方式时，每个结点存储多个字符有助于提高存储密度。若采用结点大小相同的链表存储串，则串比较、求子串、串连接、串替换等串的基本运算中，（ ）。

- A. 进行串的比较运算最不方便 B. 进行求子串运算最不方便
C. 进行串连接最不方便 D. 进行串替换最不方便

211.对于长度为 m ($m>1$) 的指定序列，通过初始为空的一个栈、一个队列后，错误的叙述是（ ）。

- A. 若入栈和入队的序列相同，则出栈序列和出队序列可能相同
B. 若入栈和入队的序列相同，则出栈序列和出队序列可以互为逆序
C. 入队序列与出队序列关系为 $1:1$ ，而入栈序列与出栈序列关系是 $1:n(n\geq 1)$
D. 入栈序列与出栈序列关系为 $1:1$ ，而入队序列与出队序列关系是 $1:n(n\geq 1)$

212.单向链表中往往含有一个头结点，该结点不存储数据元素，一般令链表的头指针指向该结点，而该结点指针域的值域为第一个元素结点的指针。以下关于单链表头结点的叙述中，错误的是（ ）。

- A. 若在头结点中存入链表长度值，则求链表长度运算的时间复杂度为 $O(1)$
B. 在链表的任何一个元素前后进行插入和删除操作可用一致的方式进行处理
C. 加入头结点后，代表链表的头指针不因为链表为空而改变
D. 加入头结点后，在链表中进行查找运算的时间复杂度为 $O(1)$

213.邻接矩阵和邻接表是图（网）的两种基本存储结构，对于具有 n 个顶点、 e 条边的图，（ ）。

- A. 进行深度优先遍历运算所消耗的时间与采用哪一种存储结构无关
B. 进行广度优先遍历运算所消耗的时间与采用哪一种存储结构无关
C. 采用邻接表表示图时，查找所有顶点的邻接顶点的时间复杂度为 $O(n*e)$
D. 采用邻接矩阵表示图时，查找所有顶点的邻接顶点的时间复杂度为 $O(n^2)$

214.已知一个二叉树的先序遍历序列为①、②、③、④、⑤，中序遍历序列为②、①、④、③、⑤，则该二叉树的后序遍历序列为（ ）。对于任意一棵二叉树，叙述错误的是（ ）。

问题 1: A. ②、③、①、⑤、④ B. ①、②、③、④、⑤ C. ②、④、⑤、③、① D. ④、⑤、③、②、①

问题 2:

- A. 由其后续遍历序列和中序遍历序列可以构造该二叉树的先序遍历序列
B. 由其先序遍历序列和后序遍历序列可以构造该二叉树的中序遍历序列
C. 由其层序遍历序列和中序遍历序列可以构造该二叉树的先序遍历序列
D. 由其层序遍历序列和后序遍历序列不能构造该二叉树的中序遍历序列