

第一章 数据结构

1.1 选择题

1. ▲ 下列程序段的时间复杂度是 ____

```
int sum = 0;
    for(int i = 1; i < n; i *= 2)
        for(int j = 0; j < i; j++)
            sum++;
```

2. 关于线性表的顺序存储和链式存储结构的描述中, 正确的是 ()

- (1) 线性表的顺序结构优于其链式存储结构
- (2) 链式存储结构比顺序存储结构能更方便地表示各种逻辑结构
- (3) 若频繁使用插入和删除操作, 则顺序存储结构更优于链式存储结构
- (4) 顺序存储结构和链式存储结构都可以进行顺序存取

A. 1,2,3

B. 2,4

C. 2,3

D. 3,4

3. 对于一个头指针为 *head* 的带头结点的单链表, 判断该表为空表的条件是 (), 对于不带头结点的单链表, 判断空表的条件是 ()

A. $head == NULL$

B. $head \rightarrow next == NULL$

C. $head \rightarrow next == head$

D. $head \neq NULL$

4. 一个链表最常用的操作为在末尾插入结点和删除节点, 则选用 () 最节省时间.

A. 带头结点的双循环链表

B. 单循环列表

C. 带尾结点的单循环链表

D. 单链表

5. 设对 $n(n > 1)$ 元素的线性表运算只有 4 种, 删除第一个元素, 删除最后一个元素, 在第一个元素之前插入一个元素, 在最后一个元素之后插入一个元素, 则最好使用 ()
- A. 只有尾结点指针没有头结点指针的循环单链表
B. 只有尾结点指针没有头结点指针的非循环双链表
C. 只有头结点指针没有尾结点指针的循环双链表
D. 既有头结点又有尾结点的循环单链表
6. 假定利用数组 $a[n]$ 存储一个栈, 初始栈顶指针 $top == -1$, 则元素 x 进栈的操作为 ()
- A. $a[-top] = x$ B. $a[top--] = x$ C. $a[++top] = x$ D. $a[top++] = x$
7. 和顺序栈相比, 链栈有一个比较明显的优势, 即 ()
- 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
8. 链栈 (不带头结点) 执行 Pop 操作, 并将出栈元素存在 x 中, 应该执行 ()
- A. $x = top; top = top \rightarrow next$ B. $x = top \rightarrow data$
C. $top = top \rightarrow next; x = top \rightarrow data$ D. $x = top \rightarrow data; top = top \rightarrow next$
9. 三个不同元素进栈, 能够得到 () 不同的出栈序列
10. 一个栈的输入序列为 $1, 2, \dots, n$ 输出序列的第一个元素为 i , 则第 j 个输出元素是 ()
- A. 不确定 B. $n - i$ C. $n - i - 1$ D. $n - i + 1$
11. 设栈的初始状态为空, 当字符序列 nl_ 作为栈的输入时, 输出长度为 3, 且可用做 C 语言标识符的序列有 () 个
- A. 3 B. 4 C. 5 D. 6
12. 设有一个顺序共享栈 $Share[0:n-1]$, 其中第一个栈顶指针 $top1$ 的初始值为 -1, 第二个栈顶指针 $top2$ 的初始值为 n , 则判断共享栈满的条件是 ()
- A. $top2 - top1 == 1$ B. $top1 - top2 == 1$
C. $top1 == top2$ D. 以上对不对

13. ◆ 若元素a,b,c,d,e,f依次进栈, 允许进栈, 出栈交替进行, 但不允许连续 3 次进栈, 退栈操作, 不可能得到的出栈序列是 ()
- A. dcebfa B. cbdaef C. bcaefd D. afedcb
14. ◆ 一个栈的入栈序列为1, 2, 3, ..., n 出栈序列为 P_1, P_2, \dots, P_n . 若 $P_2 = 3$ 则 P_3 可能的取值的个数是 ()
- A. $n-3$ B. $n-2$ C. $n-1$ D. 无法确定
15. ◆ 若栈 S1 中保存整数, 栈 S2 中保存运算符, 函数 F() 依次执行如下各步操作:
- (1) 从 S1 中依次弹出两个操作数 a 和 b
 - (2) 从 S2 中弹出一个运算符 op
 - (3) 执行相应的运算 $b \text{ op } a$
 - (4) 将运算结果压入 S1 中
- 假定 S1 中的操作数一次是5, 8, 3, 2(2 在栈顶), S2 中的运算符依次是*, -, +(栈顶). 调用 3 次 F() 后, S1 栈顶保存的值是 ()
- A. -15 B. 15 C. -20 D. 20
16. 循环队列存储在数组 $A[0 \dots n]$ 中, 其中 $rear$ 为队尾指针, $front$ 为队首指针. 则入队时的操作为 ____; 出队时的操作为 ____; 判断队空的操作为 ____; 判断队满的操作为 ____
17. 用链式存储方法的队列进行删除操作时需要 ()
- A. 仅修改头指针 B. 仅修改尾指针
- C. 头尾指针都要修改 D. 头尾指针可能都要修改
18. 假设循环单链表表示的队列长度为 n , 队头固定在链表尾, 若只设置头指针, 则进队操作的时间复杂度为 ()
- A. $O(n)$ B. $O(1)$ C. $O(n^2)$ D. $O(n \log_2 n)$
19. ◆ 已知循环队列存储在一维数组 $A[0 \dots n-1]$ 中, 且队列非空时 $front$ 和 $rear$ 分别指向队头元素和队尾元素. 若初始队列为空, 且要求第一个进入队列的元素存储在 $A[0]$ 处, 则初始时 $front$ 和 $rear$ 的值分别是 ()

- A. 0,0 B. 0, n-1 C. n-1,0 D. n-1,n-1

20. 循环队列放在一维数组 $A[0 \dots M-1]$ 中, $end1$ 指向对首元素, $end2$ 指向队尾元素的后一个位置. 假设队列两端均可进行入队与出队操作, 队列中最多能容纳 $M-1$ 个元素. 初始时空, 下列判断队满和队空的条件中, 正确的是 ()

- A. 队空: $end1 == end2$ 队满: $end1 == (end2 + 1) \bmod M$
 B. 队空: $end1 == end2$ 队满: $end2 == (end1 + 1) \bmod (M - 1)$
 C. 对空: $end2 == (end1 + 1) \bmod M$ 队满: $end1 == (end2 + 1) \bmod M$
 D. 对空: $end1 == (end2 + 1) \bmod M$ 队满: $end2 == (end1 + 1) \bmod (M - 1)$

21. ♦ 已知操作符包含 $+, -, *, /, (,)$. 将中缀表达式 $a + b - a * ((c + d) / e - f) + g$ 转换为等价的后缀表达式 (逆波兰表示法), 用栈来实现. 初始时栈为空, 转换过程中栈中至多保存 () 个操作符.

22. ▲ 有一个 $n \times n$ 的对称矩阵 A , 将其下三角部分按行存放在一维数组 B 中, 而 $A[0][0]$ 存放在 $B[0]$ 中, 则第 $i+1$ 行对角元素 $A[i][i]$ 存放在 B 中的 () 处

- A. $(i+3)i/2$ B. $(i+1)i/2$ C. $(2n-i+1)i/2$ D. $(2n-i-1)i/2$

23. ♦ 由一个 100 阶的三对角矩阵 M , 其元素 $m_{i,j} (1 \leq i, j \leq 100)$ 按行优先依次压入下标从 0 开始的一维数组 N 中. 元素 $m_{30,30}$ 在 N 中的下标是 ()

24. ▲ 在 KMP 算法中, 串 ababaaababaa 的 PM 数组, Next 数组, Nextval 数组分别为?

25. ♦ 设主串 $T = abaabaabcabaabc$ 模式串 $S = abaabc$ 采用 KMP 算法进行模式匹配, 到匹配成功为止, 在匹配过程中进行的单个字符间的比较次数是 ()

26. 树的路径长度是从树根到每个结点的路径长度的 ()

- A. 总和 B. 最小值 C. 最大值 D. 平均值

27. (判断正误)

(1) 度为 2 的有序树就是二叉树

(2) 结点按完全二叉树层序编号的二叉树中, 第 i 个结点的左孩子编号为 $2i$

28. 具有 10 个叶结点的二叉树中有 () 个度为 2 的结点

29. 设二叉树有 $2n$ 个结点, 且 $m < n$, 则不可能存在 () 的结点
- A. n 个度为 0 B. $2m$ 个度为 0 C. $2m$ 个度为 1 D. $2m$ 个度为 2
30. 已知一颗完全二叉树的第 6 层 (设根为第一层) 有 8 个结点, 则完全二叉树的结点个数最少是 ()
31. 一颗完全二叉树上有 1001 个结点, 其中叶结点的个数是 ()
32. ♦ 对于任意一颗高度为 5 且有 10 个结点的二叉树, 若采用顺序存储结构保存, 每个节点占 1 个存储单元 (仅保存结点的数据信息), 则存放该二叉树需要的存储单元数量至少是 ()
33. 在下列关于二叉树遍历的说法中, 正确的是 ()
- A. 若有一个结点是二叉树中某个子树的中序遍历结果序列的最后一个结点, 则它一定是该子树的前序遍历结果序列的最后一个结点。
- B. 若有一个结点是二叉树中某个子树的前序遍历结果序列的最后一个结点, 则它一定是该子树的中序遍历结果序列的最后一个结点。
- C. 若有一个叶结点是二叉树中某个子树的中序遍历结果序列的最后一个结点, 则它一定是该子树的前序遍历结果序列的最后一个结点。
- D. 若有一个叶结点是二叉树中某个子树的前序遍历结果序列的最后一个结点, 则它一定是该子树的中序遍历结果序列的最后一个结点。
34. 设 n, m 为一颗二叉树上的两个结点, 在后序遍历时, n 在 m 前的充分条件是 ()
- A. n 在 m 的右方 B. n 是 m 的祖先 C. n 在 m 的左方 D. n 是 m 的子孙
35. 在二叉树中的两个结点 m 和 n , 若 m 是 n 的祖先, 则使用 () 可以找到从 m 到 n 的路径
36. 若二叉树中结点的先序序列是 $\dots a \dots b \dots$, 中序序列是 $\dots b \dots a \dots$ 则 ()
- A. 结点 a 和结点 b 分别在某结点的左子树和右子树中
- B. 结点 b 和结点 a 的右孩子中
- C. 结点 b 在结点 a 的左孩子中
- D. 结点 a 和结点 b 分别在某结点的两颗分非空子树中

37. 线索二叉树是()结构

- A. 逻辑 B. 逻辑和存储 C. 物理 D. 线性

38. 一颗左子树为空的二叉树的先序线索化后, 其中空的链域的个数是()

- A. 不确定 B. 0 个 C. 1 个 D. 2 个

39. 二叉树在线索化后, 仍然不能有效求解的问题是()

- A. 先序线索二叉树求先序后继 B. 中序线索二叉树求中序后继
C. 中序线索二叉树求中序前驱 D. 后序线索二叉树求后序后继

40. 若 X 是二叉中序线索树中一个有左孩子的结点, 且 X 不为根, 则 X 的前缀为()

- A. X 的双亲 B. X 的右子树中最左节点
C. X 的左子树中最右结点 D. X 的左子树中最右的叶结点

41. () 遍历仍然需要栈的支持.

- A. 先序线索树 B. 中序线索树 C. 后序线索树 D. 所有线索树

42. ♦ 先序序列为 a,b,c,d 的不同二叉树的个数是()

43. ♦ 若结点 p 和 q 在二叉树 T 的中序遍历序列中相邻, 且 p 在 q 之前, 则下列 p 和 q 的关系中, 不可能的是()

- (1) q 是 p 的双亲
(2) q 是 p 的右孩子
(3) q 是 p 的右兄弟
(4) q 是 p 的双亲的双亲

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

44. 利用二叉链表存储森林时, 根结点的右指针是()

- A. 指向最左兄弟 B. 指向最右兄弟 C. 一定为空 D. 不一定为空

45. 森林 $T = (T_1, T_2, \dots, T_m)$ 转换为二叉树 BT 的过程为: 若 $m=0$, 则 BT 为空, 若 $m \neq 0$, 则()
()

- A. 将中间子树 $T_{mid}(mid = (1 + m)/2)$ 的根作为 BT 的根; 将 $(T_1, T_2, \dots, T_{mid-1})$ 转换为 BT 的左子树; 将 (T_{mid+1}, \dots, T_m) 转换为 BT 的右子树
- B. 将子树 T_1 的根作为 BT 的根, 将 T_1 的子树森林转换为 BT 的左子树; 将 (T_2, T_3, \dots, T_m) 转换 BT 的右子树
- C. 将子树 T_1 根作为 BT 的根, 将 T_1 的左子森林转换为 BT 的左子树; 右子森林转换右子树, 其他类似
- D. 将森林 T 的根作为 BT 的根, 将 (T_1, \dots, T_m) 转换为该根下的结点, 得到一棵树, 然后将这棵树转换为二叉树
46. 设 F 是一个森林, B 是由 F 转换为来的二叉树. 若 F 中有 n 个非终端节点, 则 B 中右指针域为空的结点数是 ()
47. ◆ 将森林转换为对应的二叉树, 若二叉树中, 结点 u 是结点 v 的父结点的父结点, 则原来的森林中, u 和 v 可能具有关系是 ()
- (1) 父子关系
- (2) 兄弟关系
- (3) u 的父结点和 v 的父结点是兄弟关系
- A. 2 B. 1,2 C. 1,3 D. 1,2,3
48. ◆ 已知一颗有 2011 个结点的树, 其叶结点个数为 116, 则该树对应的二叉树中无右孩子的结点个数为 ()
- A. 115 B. 116 C. 1895 D. 1896
49. ◆ 若森林 F 有 15 条边, 25 个结点, 则 F 包含树的个数是 ()
50. ◆ 若将一颗树 T 转换为对应的二叉树 BT, 则下列对 BT 的遍历中, 其遍历序列与 T 的后根遍历序列相同的是 ()
- A. 先序遍历 B. 中序遍历 C. 后序遍历 D. 层序遍历
51. 在有 n 个叶节点的哈夫曼树中, 非叶结点的总数是 ()
52. 设哈夫曼编码的长度不超过 4, 若已对两个字符编码为 1 和 01, 则还最多可以对 () 个个字符编码

53. 一下对于哈夫曼树的说法中, 错误的是 ()
- A. 对应一组权值构造出来的哈夫曼树一般不是唯一的
 - B. 哈夫曼树具有最小的带权路径长度
 - C. 哈夫曼树中没有度为 1 的结点
 - D. 哈夫曼树中除了度为 1 的节点外, 还有度为 2 的结点和叶结点
54. 若度为 m 的哈夫曼树中, 叶结点的数目为 n , 则非叶结点的数目为 ()
55. ♦ 已知字符集 a, b, c, d, e, f 若各字符出现的次数分别为 $6, 3, 8, 2, 10, 4$ 则对应字符集中的各字符的哈夫曼编码可能是 ()
- A. 00, 1011, 01, 1010, 11, 100
 - B. 00, 100, 110, 000, 0010, 01
 - C. 10, 1011, 11, 0011, 00, 010
 - D. 0011, 10, 11, 0010, 01, 000
56. ♦ 对应任意给定的含有 n 个字符的有限集合 S , 用二叉树表示 S 的哈夫曼编码集和定长编码集, 分别得到二叉树 T_1 和 T_2 . 下列叙述正确的是 ()
- A. T_1 和 T_2 的结点数相同
 - B. T_1 的高度大于 T_2 的高度
 - C. 出现频次不同的字符在 T_1 中处于不同的层
 - D. 出现频次不同的字符在 T_2 中处于相同的层
57. 以下关于图的叙述中, 正确的是 ()
- A. 图与树的区别在于图的边数大于等于顶点数
 - B. 假设有图 $G = \{V, \{E\}\}$, 顶点集 $V' \subseteq V, E' \subseteq E$ 则 V' 和 $\{E'\}$ 构成 G 的子图
 - C. 无向图的连通分量是指无向图的极大连通子图
 - D. 图的遍历就是从图中的某一顶点出发遍历图中的其余顶点
58. 以下关于图的说法, 正确的是 ()
- A. 强连通有向图的任何顶点到其他顶点都有弧
 - B. 图的任意顶点的入度都等于出度
 - C. 有向完全图一定是强连通有向图
 - D. 有向图的边集的子集和顶点集的子集可构成原有有向图的子图

59. 对于一个有 n 个顶点的图;若是连通无向图,其边的个数至少是 ();若是强连通有向图,其边的个数至少为 ()
60. 在有 n 个顶点的有向图中,顶点的度最大可以达到 ()
61. 设无向图 $G = (V, E), G' = (V', E')$ 若 G' 是 G 的生成树,则下列不正确的是 ()
- (1) G' 为 G 的连通分量
 - (2) G' 为 G 的无环子图
 - (3) G' 为 G 的极小连通子图且 $V' = V$
- A. 1,2 B. 3 C. 2,3 D. 1
62. ◆ 下列关于无向连通图特性的叙述中,正确的是 ()
- (1) 所有顶点的度之和为偶数
 - (2) 边数大于顶点数减一
 - (3) 至少有一个顶点的度为一
- A. 1 B. 2 C. 1,2 D. 1,3
63. 带权有向图 G 用邻接矩阵存储,则 v_i 的入度等于邻接矩阵中 ()
- A. 第 i 行非 ∞ 的元素个数 B. 第 i 列非 ∞ 的元素个数
- C. 第 i 行非 ∞ 且非 0 的元素个数 D. 第 i 列非 ∞ 且非 0 的元素个数
64. 无向图 G 中包含 $N(N>15)$ 个顶点,以邻接矩阵形式存储时共占用 N^2 个存储单元 (其他辅助空间忽略不计);以邻接表形式存储时,每个表结点占用 3 个存储单元,每个头结点占用 2 个存储单元 (其他辅助空间忽略不计).若令图 G 的邻接矩阵存储所占空间小于等于邻接表存储所占空间,该图 G 所包含的边的数量至少是 ()
65. n 个顶点的无向图的邻接表中最多有 () 个边表节点
66. 假设有 n 个顶点, e 条边的有向图用邻接表表示,则删除与某个顶点 v 相关的所有边的时间复杂度是 ()
67. 对于一个有 n 个顶点, e 条边的图采用邻接表表示时,进行 DFS 遍历的时间复杂度是 (),空间复杂度是 ();进行 BFS 遍历的时间复杂度是 (),空间复杂度是 ()

68. 对于一个有 n 个顶点, e 条边的图采用邻接矩阵表示时, 进行 DFS 遍历的时间复杂度是 (), 空间复杂度是 (); 进行 BFS 遍历的时间复杂度是 (), 空间复杂度是 ()
69. 图的广度优先生成树的树高比深度优先生成的树高 ()
- A. 小或相等 B. 小 C. 大或相等 D. 大
70. 从无向图的任意顶点出发进行一次深度优先遍历便可以访问所有顶点, 则该图一定是 ()
- A. 完全图 B. 连通图 C. 有回路 D. 一棵树
71. 一下叙述中, 正确的是 ()
- A. 最短路径一定是简单路径
- B. Dijkstra 算法不适合求有环路的带权图的最短路径
- C. Dijkstra 算法不适合求任意两个顶点的最短路径
- D. Floyd 算法求两个顶点的最短路径, $path_k - 1$ 一定是 $path_k$ 的子集
72. 若一个有向图的顶点不能排成一个拓扑序列, 则可以判断该有向图 ()
- A. 含有多个出度为 0 的顶点 B. 是一个强连通图
- C. 含有多个入度为 0 的顶点 D. 含有顶点数大于 1 的强连通分量
73. 下列关于图的说法中, 正确的是 ()
- (1) 有向图中顶点 V 的度等于其邻接矩阵中第 V 行中 1 的个数
- (2) 无向图的邻接矩阵一定是对称矩阵, 有向图的邻接矩阵一定是非对称矩阵
- (3) 在带权图 G 的最小生成树 G_i 中, 某条边的权值可能会超过为选边的权值
- (4) 若有向无环图的拓扑序列唯一, 则可以唯一确定该图
- A. 1,2,3 B. 3,4 C. 3 D. 4
74. 已知带权图为 $G = (V, E)$, 其中 $V = \{v_1, v_2, \dots, v_{10}\}$, 边集合为 $E = \{ \langle v_1, v_2 \rangle 5, \langle v_1, v_3 \rangle 6, \langle v_2, v_5 \rangle 3, \langle v_3, v_5 \rangle 6, \langle v_3, v_4 \rangle 3, \langle v_4, v_5 \rangle 3, \langle v_4, v_7 \rangle 1, \langle v_4, v_8 \rangle 4, \langle v_5, v_6 \rangle 4, \langle v_5, v_7 \rangle 2, \langle v_6, v_{10} \rangle 4, \langle v_7, v_9 \rangle 5, \langle v_8, v_9 \rangle 2, \langle v_9, v_{10} \rangle 2 \}$ 则 G 的关键路径长度为 ()

75. 下列关于关键路径的说法中, 正确的是 ()

- (1) 改变网上某一关键路径上的某一关键路径, 必将产生不同的关键路径
- (2) 在 AOE 图中, 关键路径上活动的时间延长多少, 整个工期的时间也就随之延长多少
- (3) 缩短关键路径上任意一个关键活动的持续时间可缩短关键路径长度
- (4) 缩短所有关键路径上共有的任意一个关键活动的持续时间可缩短关键路径的长度
- (5) 缩短多条关键路径上共有的任意一个关键活动的持续时间可缩短关键路径长度

A. 2,5

B. 1,2,4

C. 2,4

D. 1,4

76. ◆ 若用临接矩阵存储有向图, 矩阵中主对角线以下的元素全为零, 则关于该图拓扑序列的结论是 ()

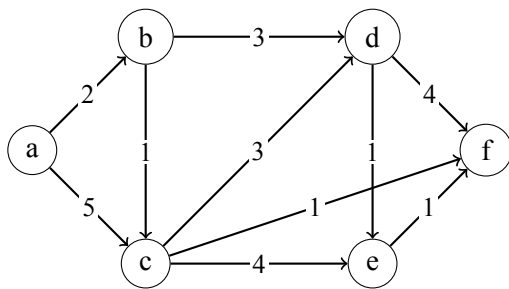
A. 存在, 且唯一

B. 存在, 且不唯一

C. 存在, 可能唯一

D. 无法确定是否存在

77. ◆ 对下列图所示的有向带权图, 若采用 Dijkstra 算法求源点 a 到其他个顶点的最短路径, 则得到的第一条最短路径的目标顶点是 b, 第二条最短路径的目标顶点是 c, 后续得到的其余各最短路径的目标顶点一次是 ()



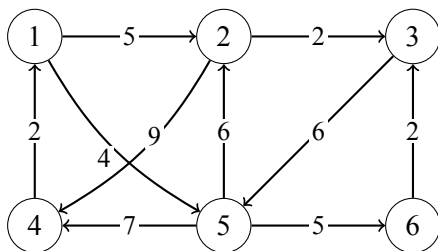
A. d,e,f

B. e,d,f

C. f,d,e

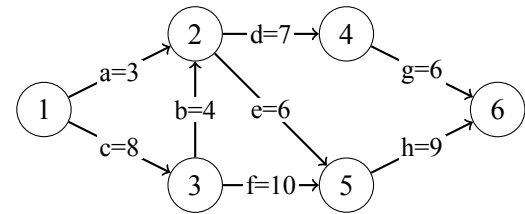
D. f,e,d

78. ◆ 使用 Dijkstra 算法求下图中从顶点 1 到其他个顶点的最短路径, 依次得到的各最短路径的目标顶点是 ()



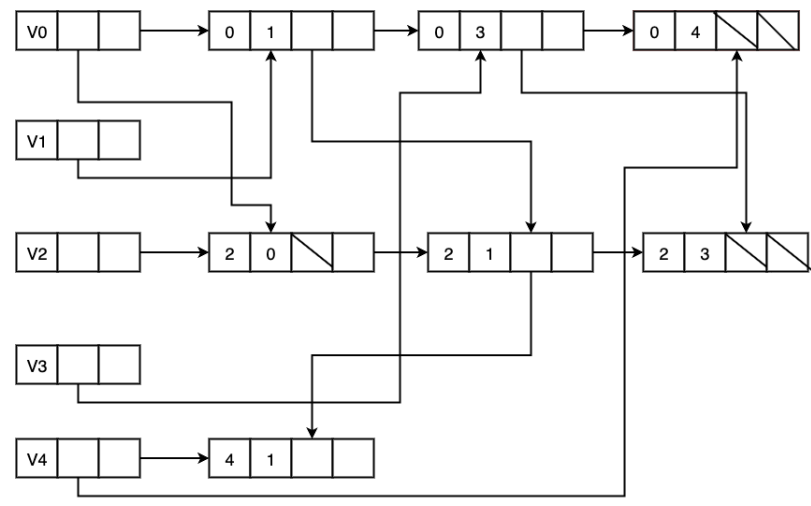
- A. 5,2,3,4,6 B. 5,2,3,6,4 C. 5,2,4,3,6 D. 5,2,6,3,4

79. ♦ 下列所示的 AOE 网表示一项包含 8 个活动的工程, 活动 d 的最早开始时间和最迟开始时间分别是 ()



- A. 3,7 B. 12,12 C. 12,14 D. 15,15

80. 图 G 利用十字链表法表示如下, 请问图 G 可能的拓扑排序为 ()



- A. V_2, V_0, V_3, V_1, V_4 B. V_0, V_3, V_1, V_4, V_2 C. V_2, V_0, V_4, V_3, V_1 D. 不存在拓扑序列

81. 以下对于最小生成树的描述, 正确的是 ()

- (1) 所有无向连通图的最小生成树一定有多个
- (2) *Prim* 和 *Kruskal* 算法构建的最小生成树一定不同
- (3) 只要无向图中不存在相同权值的边, 则该无向图的最小生成树唯一
- (4) 只要无向图中存在权值相同的边, 则该无向图的最小生成树一定不唯一
- (5) 在具有 n 个顶点的无向图 G 中, 含有 n 个顶点, $n-1$ 条边的 G 的子图就是 G 的生成树
- (6) 生成树就是最小生成树

89. 折半查找和二叉排序树的时间性能 ()

- A. 相同 B. 有时不相同 C. 完全不同 D. 无法比较

90. 对表长为 n 的有序表进行折半查找, 其判定树的高度为 ()

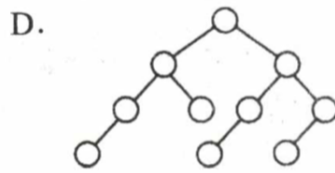
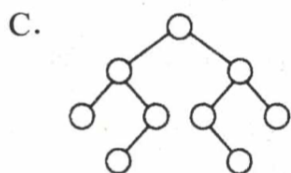
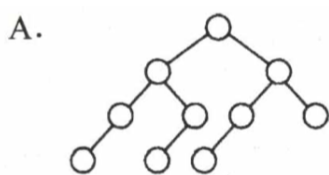
- A. $\lceil \log_2(n+1) \rceil$ B. $\log_2(n+1) - 1$ C. $\lceil \log_2 n \rceil$ D. $\lfloor \log_2 n \rfloor - 1$

91. 具有 12 个关键字的有序表中, 对每个关键字的查找概率相同, 折半查找算法查找成功的平均查找长度是 (), 折半查找失败的平均查找长度是 ()

92. 为提高查找效率, 对有 65025 个元素的有序顺序表建立索引顺序结构, 在最好的情况下查找到表中已有元素最多需要执行 () 次关键字比较

93. ◆ 已知一个长度为 16 的顺序表 L , 其元素按关键字有序排列, 若采用折半查找法查找一个 L 中不存在的元素, 则关键字的比较次数最多是 ()

94. ◆ 下列二叉树中, 可能成为折半查找判定树 (不含外部结点) 的是



95. 在含有 n 个结点的二叉排序中查找某个关键字的结点时, 最多进行 () 比较

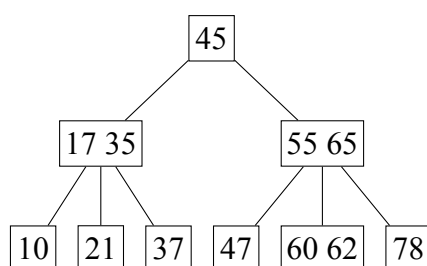
96. 构建一颗具有 n 个节点的二叉排序树时, 最理想情况下的深度为 ()

97. 含有 20 个节点的平衡二叉树的最大深度为 (), 具有 5 层结点的 AVL 树至少有 () 个结点.

98. 下列关于红黑树的说法中, 正确的是 ()

- A. 红黑树的红结点的数目最多和黑结点的数目相同
B. 若红黑树的所有结点都是黑色的, 那么它一定是一棵满二叉树
C. 红黑树的任何一个分支结点都有两个非空孩子结点
D. 红黑树的子树也一定是红黑树

99. ▲ 将关键字序列 1,2,3,4,5,6,7 一次插入初始为空的红黑树 T , 则 T 中红结点的个数是 ()
100. ◆ 现有一颗无重复关键字的平衡二叉树, 对其进行中序遍历得到一个降序序列, 下列关于该平衡二叉树的叙述中, 正确的是 ()
- A. 根结点的度一定是 2 B. 树中最小元素一定是叶结点
- C. 最后插入的元素一定是叶结点 D. 树中最大元素一定是无左子树
101. ◆ 在任意一颗非空平衡二叉树 T_1 中, 删除某结点 v 之后形成平衡二叉树 T_2 , 再将 v 插入 T_2 形成平衡二叉树 T_3 下列关于 T_1, T_3 的描述中, 正确的是 ()
- (1) 若 v 是 T_1 的叶结点, 则 T_1 和 T_3 可能不相同
- (2) 若 v 不是 T_1 的叶结点, 则 T_1 和 T_3 一定不相同
- (3) 若 v 不是 T_1 的叶节点, 则 T_1 和 T_3 一定相同
- A. 1 B. 2 C. 1,2 D. 1,3
102. 下列关于 B 与 B+ 树的描述中, 不正确的是 ()
- A. B 数和 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+ 树都可以用于文件索引结构
103. ◆ 已知一颗 3 阶 B 树, 如下图所示. 删除关键字 78 得到一颗新 B 树, 其最右叶结点中的关键字是 ()



104. ◆ 在一颗高度为 2 的 5 阶 B 树中, 所含有的关键字的个数至少是 ()
- A. 5 B. 7 C. 8 D. 14
105. ◆ 下列应用中, 适合使用 B+ 树的是 ()
- A. 编译器中的词法分析 B. 关系数据库系统的索引
- C. 网络中的路由表的快速查找 D. 操作系统的磁盘空闲块管理

106. 散列表查找成功时, 平均查找长度仅和 () 有关.

107. 在开放定址法中散列到同一地址而引起的堆积问题是由于 () 而引起的

- A. 同义词之间发生冲突 B. 非同义词之间发生冲突
C. 同义词之间或非同义词之间发生冲突 D. 散列表溢出

108. 下列关于散列冲突处理方法中, 正确的是 ()

- (1) 采用在平方探测法处理冲突时不容易产生聚集
(2) 采用线性探测法解决冲突时, 所有同义词在散列表中一定相邻
(3) 采用链地址法处理冲突时, 若限定在链首插入, 则插入任意一个元素的时间是相同的
(4) 采用链地址法处理冲突时容易引起聚集现象

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

109. 对包含 n 个元素的散列表进行查找, 平均查找长度为 ()

- A. 为 $O(\log_2 n)$ B. 为 $O(1)$ C. 不直接依赖于 n D. 直接依赖于表长 m

110. ♦ 现有长度为 11 且初始为空的散列表 HT, 散列函数 $H(k) = k \% 7$, 用线性探测再散列法解决冲突, 将关键字序列 87,40,30,6,11,22,98,20 依次插入 HT 后, HT 查找失败的平均查找长度是 ()

- A. 4 B. 5.25 C. 6 D. 6.29

111. ♦ 下列因素中, 影响哈希方法的平均查找长度是 ()

- (1) 装填因子
(2) 散列函数
(3) 冲突解决策略

- A. 1,2 B. 1,3 C. 2,3 D. 1,2,3

112. 下列关于排序的叙述中, 正确的是 ()

- A. 稳定的排序方法优于不稳定的排序方法
- B. 对同一线性表使用不同的排序方法进行排序, 得到的排序结果可能不同
- C. 排序方法都是在顺序表上实现的, 在链表上无法实现排序方法
- D. 在顺序表上实现的排序方法在链表上也可以实现
113. 对于任意 7 个关键字进行基于比较的排序, 至少要进行 () 次关键字之间的比较
- A. 13 B. 14 C. 15 D. 16
114. 用直接插入排序算法对下列 4 个表进行排序 (从小到大), 比较次数最少的是 ()
- A. 94,32,40,90,80,46,21,69 B. 21,32,46,40,80,69,90,94
- C. 32,40,21,46,69,94,90,80 D. 90,69,80,46,21,32,94,40
115. 对序列98,36,-9,0,47,23,1,8,10,7 采用希尔排序, 下列序列 () 是增量为 4 的一趟排序结果
- A. 10,7,-9,0,47,23,1,8,98,36 B. -9,0,36,98,1,8,23,47,7,10
- C. 36,98,-9,0,23,47,1,8,7,10 D. 以上都不对
116. 若用冒泡排序算法对序列10,14,26,29,41,52 从大到小进行排序, 则需要进行 () 比较
- A. 3 B. 10 C. 15 D. 25
117. 对下列关键字序列用到了快排进行排序, 速度最快的情形是 () 速度最慢的是 ()
- A. 21,25,5,17,9,23,30 B. 25,23,30,17,21,5,9
- C. 21,9,17,30,25,23,5 D. 5,9,17,21,23,25,30
118. 对于下列 4 个序列, 以第一个关键字为基准用快速排序算法进行排序, 在第一趟过程中移动记录次数最多的是 ()
- A. 92,96,88,42,30,35,110,100 B. 92,96,100,110,42,35,30,88
- C. 100,96,92,35,30,110,88,42 D. 42,30,35,92,100,96,88,110
119. 设线性表中每个元素有两个数据项 k_1, k_2 现对线性表按以下规则进行排序, 先看数据项 k_1 , 若比其值小的元素在前, 大的元素在后, 与其值相同再看 k_2 , 小的元素在前, 大的元素在后. 满足这种要求的算法是 ()

- A. 先按 k_1 进行直接插入排序, 在按 k_2 进行简单选择排序
- B. 先按 k_2 进行直接插入排序, 在按 k_1 进行简单选择排序
- C. 先按 k_1 进行简单选择排序, 在按 k_2 进行直接插入排序
- D. 先按 k_2 进行简单选择排序, 在按 k_1 进行直接插入排序
120. 若只想得到 1000 个元素组成的序列中第 10 个最小元素之前的部分排序的序列, 则用 () 方法最快.
- A. 冒泡排序 B. 快速排序 C. 希尔排序 D. 堆排序
121. 在含有 n 个关键字的小根堆中, 关键字最大的记录可能存储在 () 位置
- A. $n/2$ B. $n/2 + 2$ C. 1 D. $n/2 - 1$
122. 构建 n 个记录的初始堆, 其时间复杂度为 (), 对 n 个记录进行堆排序, 最坏情况下时间复杂度是 ()
- A. $O(n)$ B. $O(n^2)$ C. $O(\log_2 n)$ D. $O(n \log_2 n)$
123. 已知小根堆为 8,15,10,21,34,16,12 删除关键字 8 之后需要重新建堆, 关键字之间的比较次数是 ()
- A. 1 B. 2 C. 3 D. 4
124. 将序列 6,1,5,9,8,4,7 建成大根堆时, 正确的序列变化时 ()
- A. 6,1,7,9,8,4,5→6,9,7,1,8,4,5→9,6,7,1,8,4,5→9,8,7,1,6,4,5
- B. 6,9,5,1,8,4,7→6,9,7,1,8,4,5→9,6,7,1,8,4,5→9,8,7,1,6,4,5
- C. 6,9,5,1,8,4,7→9,6,5,1,8,4,7→9,6,7,1,8,4,5→9,8,7,1,6,4,5
- D. 6,1,7,9,8,4,5→7,1,6,9,8,4,5→7,9,6,1,8,4,5→9,7,6,1,8,4,5→9,8,6,1,7,4,5
125. 下列关于大根堆 (至少包含两个元素) 的叙述中, 正确的是 ()
- (1) 可以将堆视为一颗完全二叉树
- (2) 可以采用顺序存储方式保存堆
- (3) 可以将堆视为一棵二叉排序树

(4) 堆中的次大值一定在根的下一层

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

126. 若对 27 个元素值只进行三趟多路归并排序, 则选取的归并路数最少是 ()

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

127. 将两个各有 N 个元素的有序表合并为一个有序表, 最少的比较次数 (), 最多比较次数是 ()

- A. N B. $2N-1$ C. $2N$ D. $N-1$

128. 若要求排序是稳定的, 且关键字为实数, 则在下列排序中应该选用 ()

- A. 直接插入排序 B. 选择排序 C. 基数排序 D. 快速排序

129. 下列排序算法中属于稳定排序的是 (), 平均时间复杂度为 $O(n \log n)$ 的是 (), 在最好的情况下, 时间复杂度可以达到线性的时间有 ()

- A. 冒泡排序 B. 堆排序 C. 选择排序 D. 直接插入排序
E. 希尔排序 F. 归并排序 G. 快速排序

130. 若序列的原始状态为 1,2,3,4,5,10,6,7,8,9 要想使得排序过程中元素比较次数最少, 则应该采用的是 ()

- A. 插入排序 B. 选择排序 C. 希尔排序 D. 冒泡排序

131. ♦ 下列排序方法中, 若将顺序存储转换为链式存储, 则算法时间效率会降低的是 ()

- A. 插入排序 B. 选择排序 C. 冒泡排序
D. 希尔排序 E. 堆排序

132. 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, 若不采用败者树, 使用传统的顺序选择出最小记录 (简单选择排序) 的方法, 总的比较次数为 (); 若采用败者树最小的方法, 总的比较次数约为 ()

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

133. 在做 m 路平衡归并排序过程中, 为实现输入/内部归并/输出的并行处理, 需要设置 () 个输入缓冲区和 () 输出缓冲区.
- A. 2 B. m C. $2m-1$ D. $2m$
134. ♦ 已知三叉树 T 中的 6 个叶结点的权分别是 2,3,4,5,6,7, T 的带权路径长度最小是 ()
- A. 27 B. 46 C. 54 D. 56
135. ♦ 设外存上有 120 个初始归并段, 进行 12 路归并时, 为实现最佳归并, 则需要补充的虚段个数是 ()
- A. 1 B. 2 C. 3 D. 4

1.2 综合题