



1. 数据结构被形式的定义为 (K, R) ，其中 K 是 () 的有限集合， R 是 K 上关系的有限集合。

- A. 算法 B. 数据元素
C. 数据操作 D. 逻辑结构

答案: B

2. 数据元素是数据的最小单位。 ()

答案: 错误

数据元素是数据的基本单位，数据项是数据的最小单位。

3. 存储数据时，通常不仅需要存储各数据元

素的值，而且还要存储 ()。

- 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

答案: C

4. 逻辑上可以将数据结构分为 ()。

- A. 动态结构和静态结构
B. 线性结构和非线性结构
C. 顺序结构和链式结构
D. 初等结构和组合结构

答案: B

抓住关键词“逻辑”

5. 按数据元素的逻辑关系来说，数据结构可分为四种：线性表、集合、树和图，其中树形结构中的数据元素之间存在“_____”的关系。

答案: 一对多

6. 有向图是一种非线性结构。 ()

答案: 正确

7. 以下属于逻辑结构的是 ()。

- A. 顺序表 B. 哈希表
C. 有序表 D. 单链表

答案: C

一般带“顺序”、“哈希”、“链”，就是存储结构。

8. 以下是4个算法的时间复杂度函数表达式，其中时间复杂度最小的是 ()。

- A. $T(n) = 2n^3 + 3n^2 + 1000$
B. $T(n) = n^3 - 2000$
C. $T(n) = n^2 \log_2 n + n^2$
D. $T(n) = n^2 + 1000$

答案: D

根据加法规则，四个选项的时间复杂度分别是 $O(n^3)$, $O(n^3)$, $O(n^2 \log_2 n)$, $O(n^2)$

9. 算法是对特定问题求解步骤的一种描述，它具有_____、_____、可行性、输入和输出五个重要的特性。

答案: 有穷性 确定性

10. 求下列程序段的时间复杂度。

```
(1) for(i=0; i<=m; i++)
    for(j=0; j<n; j++)
        A[i][j]=0;
```

```
(2) y=0;
    while((y+1)*(y+1)<=n)
        y=y+1;
```

```
(3) i=1;
    while(i<=n)
        i=i*3;
```

答案: (1) $O(mn)$

(2) $O\left(n^{\frac{1}{2}}\right)$

设循环体共执行 t 次，每循环依次，循环变量 y 加1，最终 $t = y$ 。故 $t^2 \leq n$ ，得 $T(n) = O(n^{1/2})$

(3) $O(\log_3 n)$



1. 线性表的特点是每个元素都有一个前驱和一个后继。 ()

答案: 错误

除第一个元素外, 每个元素有且仅有一个直接前驱。

除最后一个元素外, 每个元素有且仅有一个直接后继。

2. 线性表的顺序存储结构是一种 () 的存储结构。

A. 随机存取 B. 顺序存取

C. 索引存取 D. 散列存取

答案: A

3. 下述 () 是顺序存储结构的优点。

A. 存储密度大 B. 插入运算方便

C. 删除运算方便 D. 无需大片连续存储空间

答案: A

4. 两个有序顺序表分别是具有 n 个元素与 m 个元素且 $n \leq m$, 现将其归并成一个有序表, 其最少的比较次数是 ()。

A. n B. m

C. $n-1$ D. $m+n$

答案: A

这时第一个有序表 (长度为 n) 的最后一个值即最大值小于第二个有序表 (长度为 m) 的第一个值即最小值, 此时比较次数为 n 。

5. 删除顺序表中第1个数据元素 a_0 的时间

复杂度是 $O(n)$ 。 ()

答案: 正确

若有 n 个元素的话, 此时, 移动了 $n-1$ 个元素。

6. 在 n 个结点的顺序表中, 算法的时间复杂度是 $O(1)$ 的操作是 ()。

A. 访问第 i 个结点 ($1 \leq i \leq n$) 和求第 i 个结点的直接前驱 ($2 \leq i \leq n$)

B. 在第 i 个结点后插入一个新结点 ($1 \leq i \leq n$)

C. 删除第 i 个结点 ($1 \leq i \leq n$)

D. 将 n 个结点从小到大排序

答案: A

A 选项访问第 i 个结点 ($1 \leq i \leq n$), 也就是 $L.data[i-1]$

访问第 i 个结点的直接前驱 ($2 \leq i \leq n$), 也就是 $L.data[i-2]$

时间复杂度都为 $O(1)$

B 选项插入新结点需要移动 $n-i+1$ 个元素, 时间复杂度为 $O(n)$

C 选项删除一个结点需要移动 $n-i$ 个元素, 时间复杂度为 $O(n)$

D 选项排序没有给出排序方法, 所以时间复杂度不确定

7. 下面 () 不是线性表的特性。

A. 除第一个元素外, 每一个元素都有前驱

B. 除最后一个元素外, 每一个元素都有后继

C. 线性表是数据元素的有限序列

D. 线性表的长度等于 n , 并且 n 不等于 0

答案: D

线性表是具有相同数据类型的 $n (n \geq 0)$ 个数据元素的有限序列。

8. 二维数组 $A[10][20]$ 按行优先顺序存储, 每个元素占 4 个存储单元, $A[1][1]$ 的存储地址是 1000, $A[5][6]$ 的存储地址是 _____。

答案: 1340

$$1000 + ((5-1) \times 20 + (6-1)) \times 4 = 1340$$

9. 已知长度为 n 的线性表采用顺序存储结构。写一算法, 删除线性表中所有值为 x 的元素。请先对算法思想作简要文字说明, 再写出算法具体代码。

解法一: 用 k 记录顺序表 L 中等于 x 的元素个数, 边扫描 L 边统计 k , 并将不等于 x 的元素前移 k 个位置, 最后修改 L 的长度。

```
void del_x_1(SqList &L, ElemType x) {
    int k=0; i=0;
    while(i<L.length) {
        if(L.data[i]==x)
            k++;
        else
            L.data[i-k]=L.data[i];
        i++;
    }
    L.length=L.length-k;
}
```

解法二: 用 k 记录顺序表 L 中不等于 x 的元素个数 (即需要保存的元素个数), 边扫描 L 边统计 k , 并将不等于 x 的元素向前移动 k 个位置, 最后修改 L 的长度。

```
void del_x_2(SqList &L, ElemType x) {
    //本算法实现删除顺序表 L 中所有值为 x 的数据元素
    int k=0;
    for(i=0; i<L.length; i++) {
        if(L.data[i]!=x) {
            L.data[k]=L.data[i];
            k++;
        }
    }
    L.length=k;
}
```

10. 设计一个高效算法, 将顺序表 L 的所有元素逆置。

算法思想: 扫描顺序表 L 的前半部分元素, 对于元素 $L.data[i] (0 \leq i \leq L.length/2)$, 将其与后半部分的对应元素 $L.data[L.length-i-1]$ 进行交换。

```
void Reverse(SqList &L) {
    ElemType temp;
    for(i=0; i<L.length/2; i++) {
        temp=L.data[i];
        L.data[i]=L.data[L.length-i-1];
        L.data[L.length-i-1]=temp;
    }
}
```


1. 线性表采用链式存储时，其地址（ ）。

- A. 必须是连续的
- B. 一定是不连续的
- C. 部分地址必须是连续的
- D. 连续与否都可以

答案：D

2. 顺序存储方式插入和删除数据元素效率太

低，因此它不如链式存储方式好。（ ）

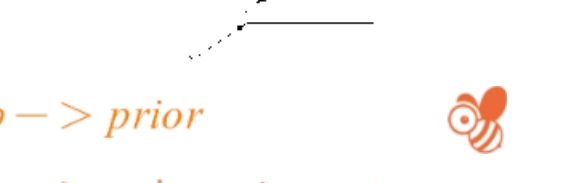
答案：错误

3. 在双向链表中，若要求在 p 指针所指的结点之

前插入指针为 s 所指的结点，则需执行下列语句：

$s \rightarrow next = p; s \rightarrow prior = \underline{\hspace{2cm}};$

$\quad \quad \quad p \rightarrow prior = s;$



答案： $p \rightarrow prior$

$p \rightarrow prior \rightarrow next$

4. 链表不具有的特点是（ ）。

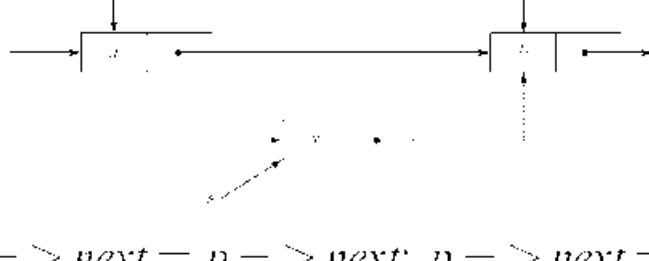
- A. 插入不需要移动元素
- B. 可随机访问任一元素
- C. 不必事先估计存储空间
- D. 删除不需要移动元素

答案：B

5. 在一个单链表中，已知 q 结点是 p 结点的前

驱结点，若在 q 和 p 之间插入 s 结点，则执行

（ ）。



- A. $s \rightarrow next = p \rightarrow next; p \rightarrow next = s;$
- B. $p \rightarrow next = s \rightarrow next; s \rightarrow next = p;$
- C. $q \rightarrow next = s; s \rightarrow next = p;$
- D. $p \rightarrow next = s; s \rightarrow next = q;$

答案：C

6. 线性表的每个结点只能是一个简单类型，而链

表的每个结点可以是一个复杂类型。（ ）

答案：错误

7. 将长度为 n 的单链表 A 链接在长度为 m 的单链

表 B 之后的算法时间复杂度为_____。

答案： $O(m)$

链接在 B 之后，需要对 B 遍历，长度为 m ，

因此遍历了 m 次找到尾结点。

8. 对于在表的首、尾两端进行插入操作的线性

表，宜采用的存储结构是（ ）。

- A. 顺序表
- B. 用头指针表示的单循环链表
- C. 用尾指针表示的单循环链表
- D. 单链表

答案：C

A 选项顺序表插入效率低，时间复杂度为

$O(n)$ ；

B，D 选项只适合在表的首端进行插入操作，

时间复杂度为 $O(1)$ ；在尾端插入时，需要遍历整

张表找到尾结点，时间复杂度为 $O(n)$ ；

C 选项在表的首、尾两端进行插入操作，时

间复杂度都为 $O(1)$ ；

9. 链表的删除算法很简单，因为当删除链中某个

结点后，计算机会自动地将后续的各个单元向前

移动。（ ）

答案：错误

链表的删除不会发生结点的移动，顺序表的

删除才会发生结点的移动。

10. 阅读下列算法，并补充所缺语句。

从头指针为 la 的带表头结点的有序顺序表中

删除所有值相同的多余元素，并释放被删除结

点的空间。

```
void purge_linkst(ListLink &la) {
    ListNode *p,*q,*t;
    ElemType temp;
    p=la->link;
    while(p!=NULL) {
        q=p;
        temp=p->data;
        p=p->link;
        if(p!=NULL&&_____ )
            p=p->link;
        else {
            while(p!=NULL&&_____) {
                t=p;
                p=p->link;
                free(t);
            }
            p=q->link;
        }
    }
}
```

答案： $temp == p \rightarrow data$

$temp == p \rightarrow data$

第一次循环用 $temp$ 保存第一个结点的值，

用 q 保存当前 p 指向的结点。

然后，用 p 指针向后遍历，若当前 p 指针

所指向结点的值与 $temp$ 相等，则删除该结点；

继续向后遍历，直到不存在这样的结点或

者 p 指针已经为空就结束该次遍历。

接着， $p = q \rightarrow link$ 使得 p 指针指向原来

p 指针的下一个结点，重复操作。



1. 栈的 *push* 和 *pop* 操作均在 () 进行。

- A. 栈顶位置
- B. 栈底位置
- C. 任意位置
- D. 中间位置

答案: A

2. 栈又称先进先出的线性表。 ()

答案: 错误

先进后出

3. 若已知一个栈的入栈序列是 $1, 2, 3, \dots, n$, 其输出序列为 $P_1, P_2, P_3, \dots, P_n$, 若 $P_1 = n$, 则 P_i 为 ()。

- A. i
- B. $n - i$
- C. $n - i + 1$
- D. 不确定

答案: C

4. 由两个栈共享一个向量空间的好处是 ()。

- A. 减少存取时间, 降低下溢发生的概率
- B. 节省存储空间, 降低上溢发生的概率
- C. 减少存取时间, 降低上溢发生的概率
- D. 节省存储空间, 降低下溢发生的概率

答案: B

5. 在循环顺序队列中, 假设以设置一个计数变量 *num* 的方法来区分队列判满和判空的条件, *front* 和 *rear* 分别为队首和队尾指针, 它们分别指向队首元素和队尾元素的下一个存储单元。队列的最大存储容量为 *MaxSize*, 则下面不是队列判满或判空条件是 ()。

- A. $front == rear$
- B. $front == rear \ \&\& \ num == 0$
- C. $front == rear \ \&\& \ num > 0$
- D. $num == MaxSize$

答案: A

队空或队满都有 $front == rear$, 因此不能作为判断条件。

6. 设栈 *S* 和队列 *Q* 的初始状态为空, 元素

$e_1, e_2, e_3, e_4, e_5, e_6$ 依次通过栈 *Q*, 一个元素进栈后即进队列 *Q*, 若六个元素出栈的序列是 $e_2, e_4, e_3, e_6, e_5, e_1$, 则栈 *S* 的容量至少应该是 _____。

答案: 3

7. 在具有 *n* 个单元的循环队列中, 队满时共

有 _____ 个元素。

答案: $n - 1$

8. 栈和队列的共同点是 ()。

- A. 都是先进先出
- B. 都是先进后出
- C. 只允许在端点处插入和删除元素
- D. 没有共同点

答案: C

9. 解决括号匹配问题, 最适合使用 () 数据结构。

- A. 堆
- B. 栈
- C. 队列
- D. 二叉树

答案: B



1. 下面关于串的叙述中, () 是不正确的。

- A. 空串是由空格构成的串
- B. 串是字符的有限序列
- C. 模式匹配是串的一种重要运算
- D. 串既可以采用顺序存储, 也可以采用链式存储

答案: A 

空格串是由空格构成的串


2. 串的长度指的是 ()。

- 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

答案: B 

3. 两个串相等必有串长度相等且 ()。

- A. 串的各位置字符任意
- B. 串中各位置字符均对应相等
- C. 两个串含有不同的字符
- D. 两个所含字符任意

答案: B 


4. 字符串 $t = 'a \text{ very cute child}'$, $s = 'I \text{ like coffee and cake}'$, 请写出下列函数的结果:

$StrLength(s) = \underline{\hspace{2cm}}$;
 $Concat(SubString(\&t, s, 3, 15), SubString(\&t, t, 12, 6)) = \underline{\hspace{2cm}}$ 。

答案: 22; like coffee and child 

5. 与线性表相比, 串的插入和删除操作的特点是 ()。

- A. 通常以串整体作为操作对象
- B. 需要更多的辅助空间
- C. 算法的时间复杂度较高
- D. 涉及移动的元素更多

答案: A 

6. 设有两个串 p 和 q , 其中 q 是 p 的子串, 求 q 在 p 中首次出现的位置的算法称为 。

答案: 模式匹配 

7. 若将 n 阶上三角矩阵 A 按列优先顺序压缩

存放在一维数组 B 中, 第一个非零元素 a_{11} 存放在 $B[0]$ 中, 则应放在 $B[k]$ 中的非零元素 a_{ij} 的下标 i, j 与 k 的对应关系为 ()。

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

答案: D 

$1 + 2 + \dots + j - 1 + i - 1$

8. 设有一个 10 阶的对称矩阵 $A[10][10]$, 采用压缩存储方式按行将矩阵中下三角部分的元素存入一维数组中, $A[0][0]$ 存入 $B[0]$ 中, 则 $A[8][5]$ 在 B 中 () 位置。


- A. 32
- B. 33
- C. 41
- D. 65

答案: C 

$1 + 2 + \dots + 7 + 8 + 6 - 1 = 41$


9. 采用带辅助向量的三元组形式实现稀疏矩阵的转置运算, 主要是为了 ()。

- A. 增加时间复杂度
- B. 提高算法效率
- C. 降低空间复杂度
- D. 节省存储空间

答案: D 

10. 下列 () 是稀疏矩阵的一种压缩存储方法。

- A. 顺序表
- B. 单链表
- C. 双向链表
- D. 三元组的顺序表

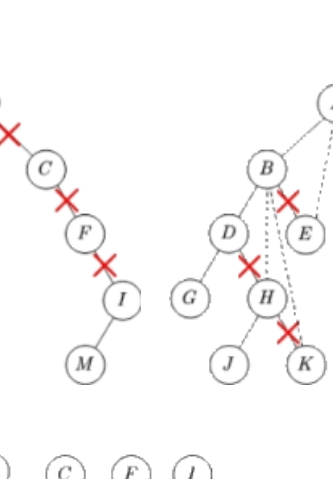
答案: D 

1. 由树转换成二叉树，其根的右孩子指针总是空的。

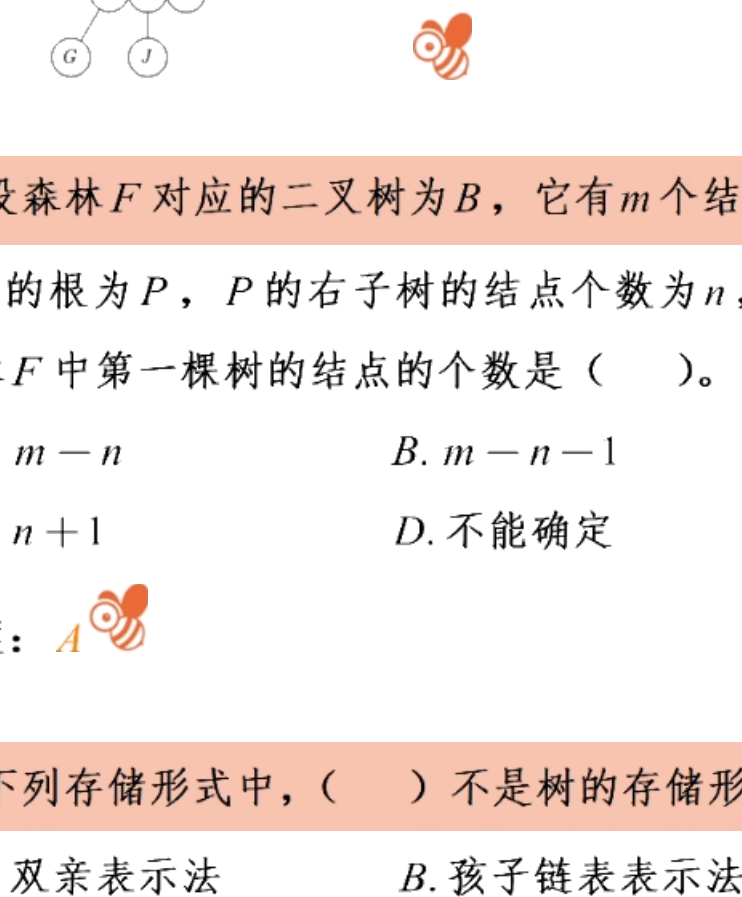
答案：正确

滚动鼠标轴或单击，开始截长图

2. 画出下列二叉树对应的森林。



答案：



3. 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 n ，森林 F 中第一棵树的结点的个数是（ ）。

- A. $m - n$ B. $m - n - 1$
C. $n + 1$ D. 不能确定

答案：A

4. 下列存储形式中，（ ）不是树的存储形式。

- A. 双亲表示法 B. 孩子链表表示法
C. 孩子兄弟表示法 D. 顺序存储表示法

答案：D

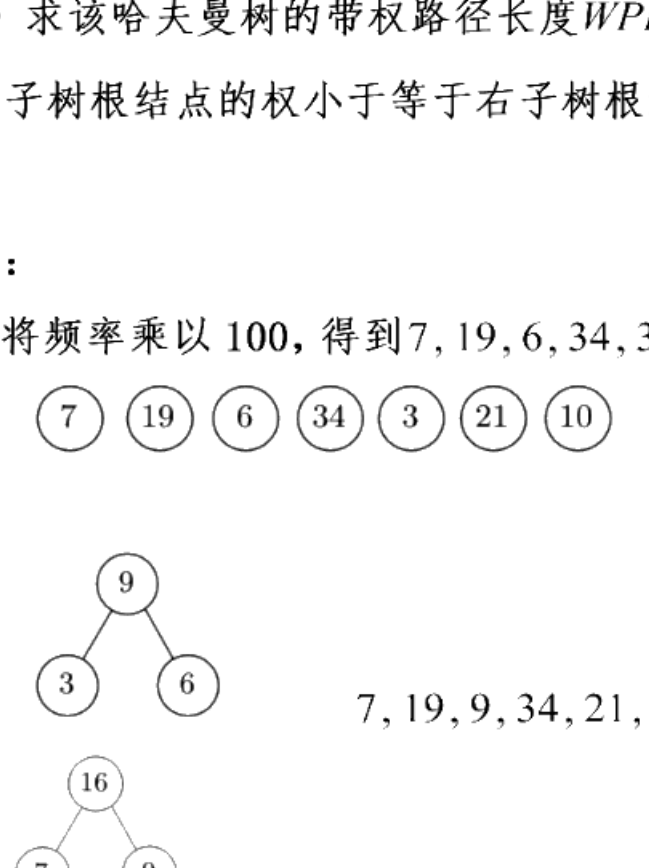
5. 如果 F 是由有序树 T 转换而来的二叉树，那么 T 中结点的后根序列就是 F 中结点的（ ）序

- A. 前序 B. 中序
C. 后序 D. 层次

答案：B

6. 假设一棵二叉树的层次序列（按层次递增顺序排列，同一层次自左向右）为 $ABECFGDHI$ ，中序序列为 $BCDAFEHIG$ 。请画出该二叉树，并将其转换成对应的森林。

答案：



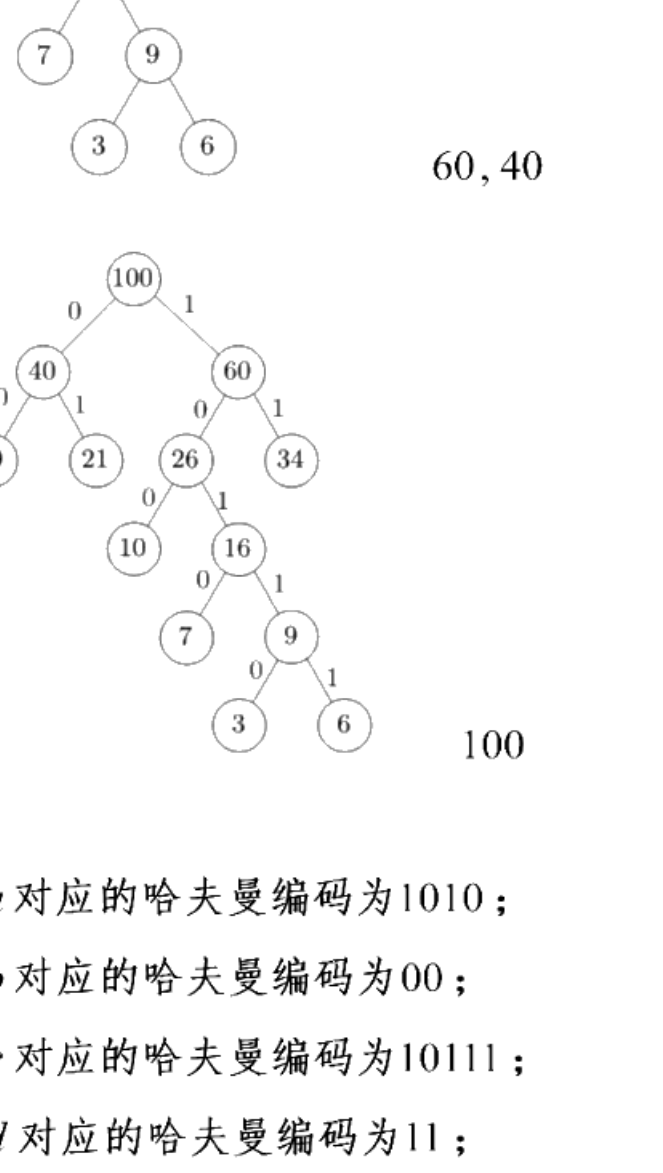
7. 设用于通信的电文仅由 7 个字母 a, b, c, d, e, f, g 组成，字母在电文中出现的频率分别是 0.07, 0.19, 0.06, 0.34, 0.03, 0.21, 0.10

(1) 请为这 7 个字母构建 *Huffman* 树，写出每个字母的 *Huffman* 编码。

(2) 求该哈夫曼树的带权路径长度 WPL 。（要求左子树根结点的权小于等于右子树根结点的权）

答案：

(1) 将频率乘以 100，得到 7, 19, 6, 34, 3, 21, 10



a 对应的哈夫曼编码为 1010；

b 对应的哈夫曼编码为 00；

c 对应的哈夫曼编码为 10111；

d 对应的哈夫曼编码为 11；

e 对应的哈夫曼编码为 10110；

f 对应的哈夫曼编码为 01；

g 对应的哈夫曼编码为 100；

(2)

$$WPL = 19 \times 2 + 21 \times 2 + 10 \times 3 + 7 \times 4 + 3 \times 5 + 6 \times 5 + 34 \times 2 = 251$$

8. 依次输入一个关键字序列

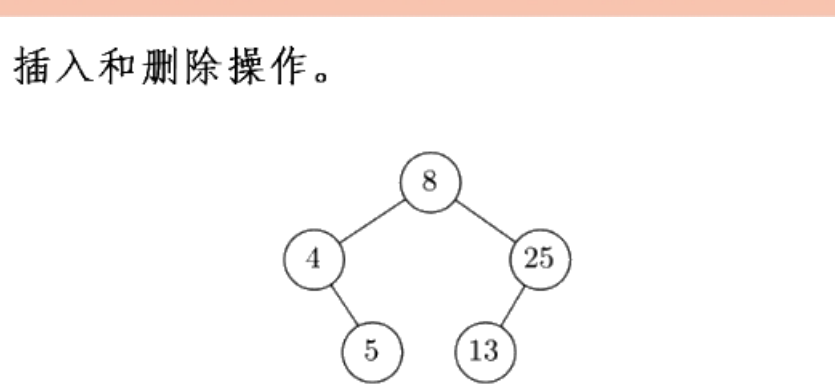
{60, 27, 76, 66, 80, 22, 70, 62}.

(1) 按输入次序构造并画出此二叉排序树；

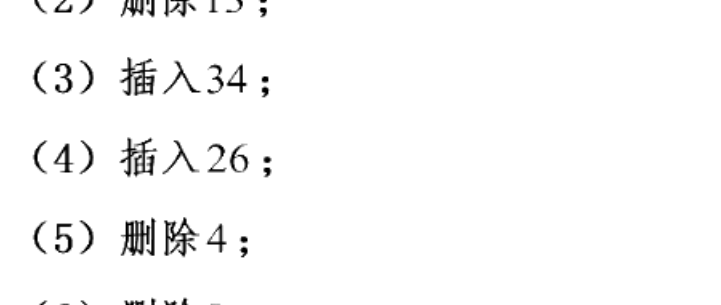
(2) 画出该树在删除关键字“76”后的二叉排序树。

答案：

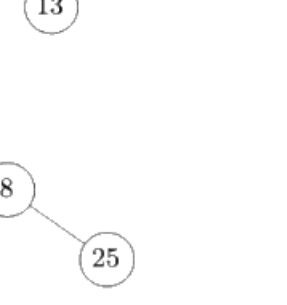
(1)



(2)



9. 如下图所示是一棵二叉排序树，现对它做如下插入和删除操作。



(1) 插入 2；

(2) 删除 13；

(3) 插入 34；

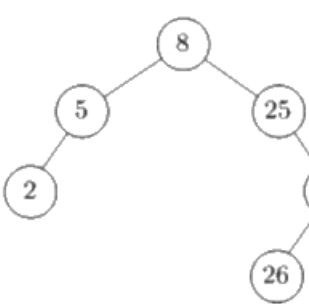
(4) 插入 26；

(5) 删除 4；

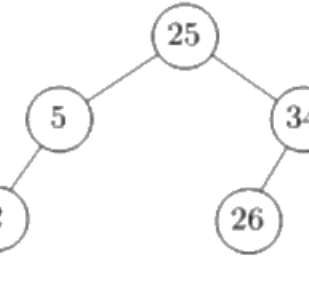
(6) 删除 8；

答案：

(1)



(2)



(3)



(4)

(5) 答案不唯一

(6) 答案不唯一

1. $n(n > 0)$ 个顶点的无向图最多有_____

条边, 最少有_____条边。

答案: $n(n-1)/2, n-1$

2. 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_____倍。

答案: 1

3. 下列关于无向连通图特性的叙述中, 正确

的是 ()。

(1) 所有顶点的度数之和为偶数

(2) 边数小于顶点个数减 1

(3) 至多有一个顶点的度数为 1

A. (1)

B. (2)

C. (1) (2)

D. (1) (3)

答案: A

4. 下面关于图的存储结构中正确的是 ()。

A. 用邻接表法存储图, 占用的存储空间大小只

与图中边数有关, 而与结点个数无关

B. 用邻接表法存储图, 占用的存储空间大小与

图中边数和结点个数都有关

C. 用邻接矩阵法存储图, 占用的存储空间大小

与图中边数和结点个数都有关

D. 用邻接矩阵法存储图, 占用的存储空间大小

只与图中边数有关, 而与结点个数无关

答案: B

用邻接表法存储图, 占用的存储空间大小

与图中边数和结点个数都有关

用邻接矩阵法存储图, 占用的存储空间大

小只与结点个数有关, 而与图中边数无关

5. 邻接矩阵适用于稀疏图的存储, 邻接表适用于

稠密图的存储。()

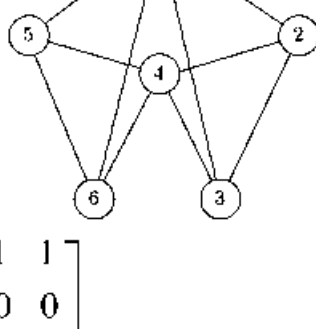
答案: 错误

邻接表适用于稀疏图的存储, 邻接矩阵适

用于稠密图的存储

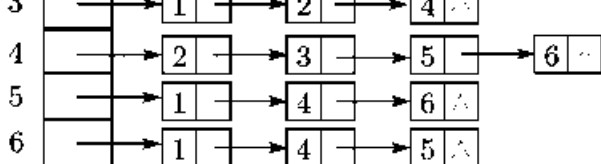
6. 如下图所示一个无向图, 试分别给出它的邻接

矩阵和邻接表。



答案:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

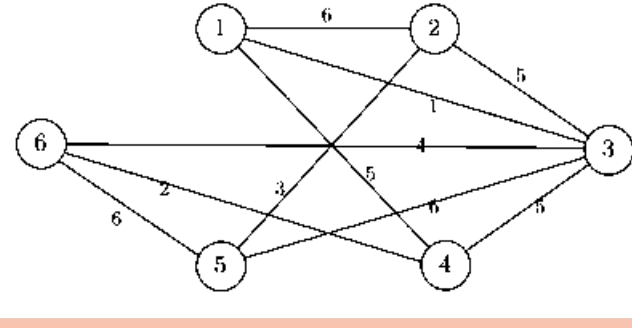


7. 如图所示是一个无向连通网的邻接矩阵,

试画出该网。

$$\begin{bmatrix} 0 & 6 & 1 & 5 & \infty & \infty \\ 6 & 0 & 5 & \infty & 3 & \infty \\ 1 & 5 & 0 & 5 & 6 & 4 \\ 5 & \infty & 5 & 0 & \infty & 2 \\ \infty & 3 & 6 & \infty & 0 & 6 \\ \infty & \infty & 4 & 2 & 6 & 0 \end{bmatrix}$$

答案:



8. 对任意一个图, 从它的某个顶点出发进行

一次深度优先或广度优先搜索遍历可访问到

该图的各个顶点。()

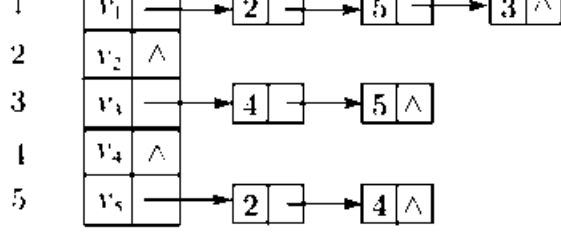
答案: 错误

该图必须是连通图

9. 已知有向图的邻接表存储结构如下图所示,

则按深度优先遍历算法从顶点 v_1 出发, 所得到的

顶点序列为 ()。



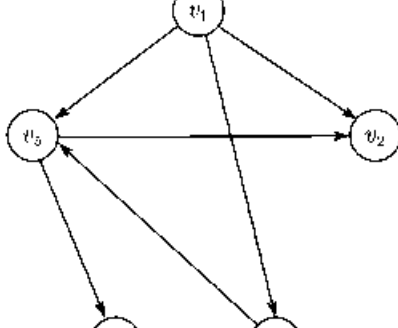
A. v_1, v_5, v_3, v_4, v_2

B. v_1, v_3, v_2, v_5, v_4

C. v_1, v_3, v_4, v_5, v_2

D. v_1, v_3, v_2, v_4, v_5

答案: C



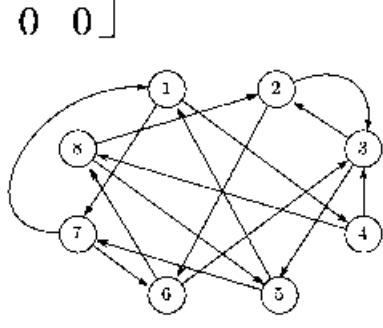
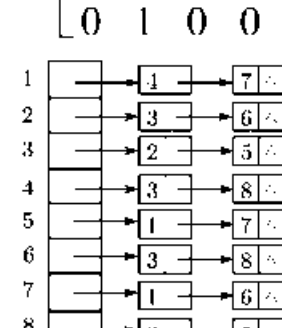
10. 已知图的邻接矩阵如下: 试给出邻接表结构,

并给出从顶点 v_1 出发进行深度优先和广度优

先的搜索结果。

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

答案:



深度优先搜索序列: 14326857

广度优先搜索序列: 14738625

1. 任何一个带权的无向连通图的最小生成树

()。

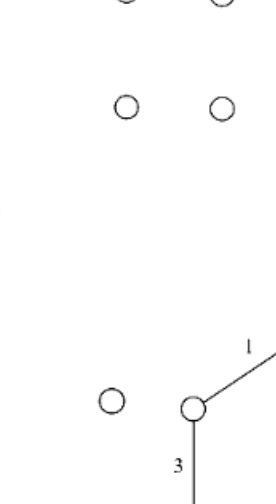
A. 只有一棵

B. 有一棵或多棵

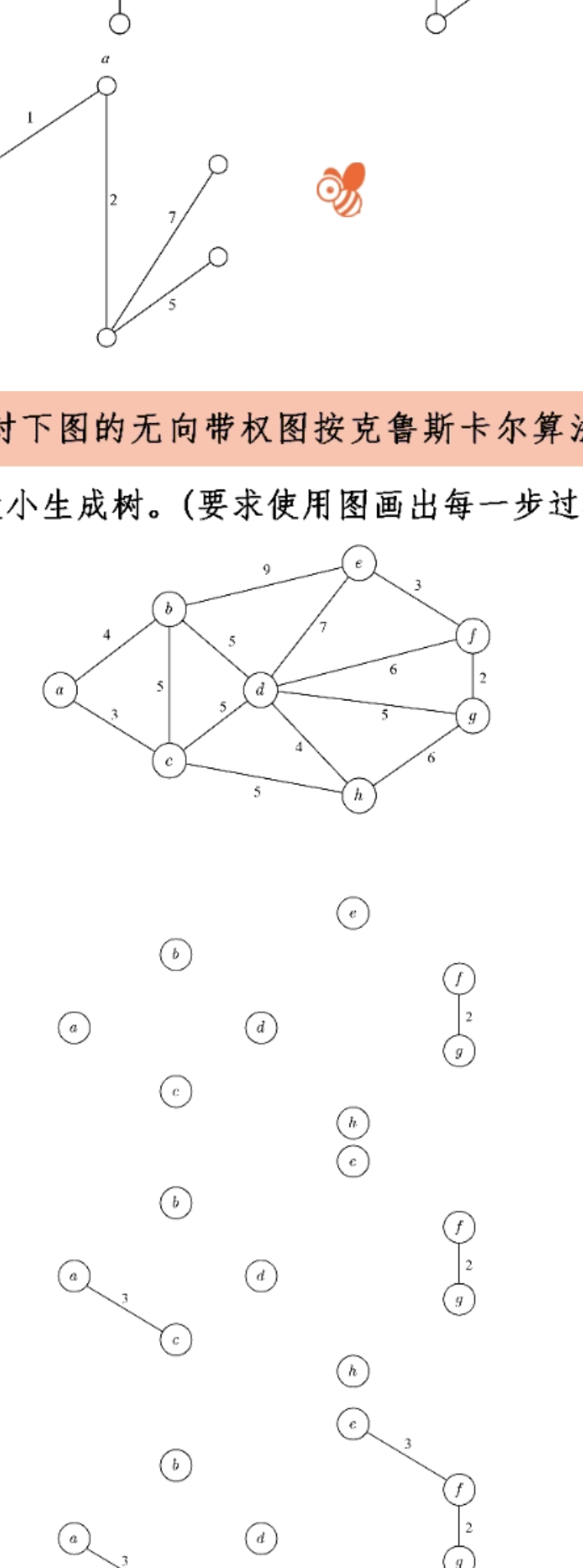
C. 一定有多棵

D. 可能不存在

答案: B

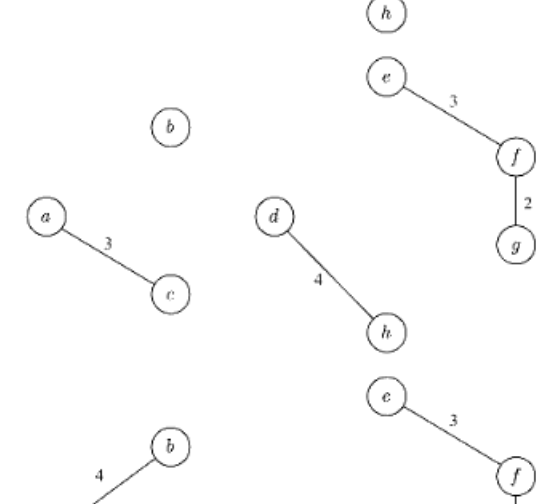
2. 一个赋权网络如下图所示。从顶点 a 开始, 用 $Prim$ 算法求出一棵最小生成树。

答案:



3. 请对下图的无向带权图按克鲁斯卡尔算法求

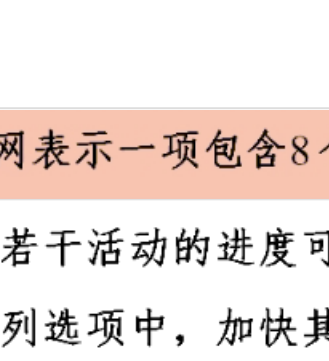
其最小生成树。(要求使用图画出每一步过程)。



答案:

4. 用 $dijkstra$ 标号法求下图中从 v_1 到其余顶点的

最短路径和距离。



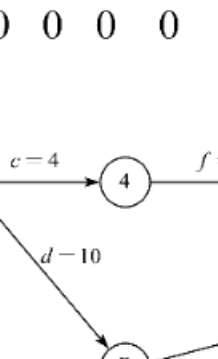
答案:

	1	2	3	4	5	6	7	8
v_1	0							
v_2	∞	6	6	6				
v_3	∞	3						
v_4	∞	∞	5					
v_5	∞	∞	8	6	6			
v_6	∞	∞	∞	∞	12	12	12	12
v_7	∞	∞	11	11	11	7		
v_8	∞	∞	∞	11	11	11	10	
	0	3	5	6	6	7	10	12
	v_1	v_3	v_4	v_2	v_5	v_7	v_8	v_6

从 v_1 到其余各点的最短路径和距离如下:

$v_1 v_3$	$d(v_1, v_3) = 3$
$v_1 v_3 v_4$	$d(v_1, v_4) = 5$
$v_1 v_2$	$d(v_1, v_2) = 6$
$v_1 v_3 v_4 v_5$	$d(v_1, v_5) = 6$
$v_1 v_3 v_4 v_5 v_7$	$d(v_1, v_7) = 7$
$v_1 v_3 v_4 v_5 v_7 v_8$	$d(v_1, v_8) = 10$
$v_1 v_2 v_6$	$d(v_1, v_6) = 12$

5. 已给下图, () 是该图的拓扑排序序列。



A. 1, 2, 3, 4

B. 1, 3, 2, 4

C. 1, 2, 4, 3

D. 1, 4, 2, 3

答案: A

6. 为判别有向图是否存在回路, 可利用 ()

算法。

A. 最短路径

B. 深度优先遍历

C. 拓扑排序

D. 最小生成树

答案: C

7. 下列关于 AOE 网的叙述中, 不正确的是

()。

A. 关键活动不按期完成就会影响整个工程的完成时间。

B. 任何一个关键活动提前完成, 那么整个工程将会提前完成。

C. 所有的关键活动提前完成, 那么整个工程将会提前完成。

D. 某些关键活动提前完成, 那么整个工程将会提前完成。

答案: B

8. 下列 AOE 网表示一项包含 8 个活动的工程, 通过同时加快若干活动的进度可以缩短整个工程的工期, 下列选项中, 加快其进度就可以缩短工程工期的是 ()。A. c 和 e B. d 和 e C. f 和 d D. f 和 h

答案: C

9. 已知一个 AOE 网采用邻接矩阵存储, 其邻接矩阵的三元组表示为 $(1, 2, 1), (1, 3, 3), (2, 4, 4), (2, 5, 10), (3, 5, 3), (4, 6, 6), (5, 7, 6), (5, 8, 7), (6, 8, 5), (7, 8, 6)$ 。(1) 画出该 AOE 网。

(2) 分别求出各事件的最早和最迟发生时间。

(3) 分别求出各活动的最早和最迟发生时间。

(4) 求出关键活动。

答案:

	0	1	3	0	0	0	0	0
	0	0	0	4	10	0	0	0
	0	0	0	0	3	0	0	0
	0	0	0	0	0	6	0	0
	0	0	0	0	0	0	6	7
	0	0	0	0	0	0	0	5
	0	0	0	0	0	0	0	6
	0	0	0	0	0	0	0	0

(1)

(2)

	1	2	3	4	5	6	7	8
$ve(i)$	0	1	3	5	11	11	17	23
$vl(i)$	0	1	8	12	11	18	17	23

(3)

	a	b	c	d	e	f	g	h	i	j
$e(i)$	0	0	1	1	3	5	11	11	11	17
$l(i)$	0	5	8	1	8	12	11	16	18	17
$l(i) - e(i)$	0	5	7	0	5	7	0	5	7	0

(4) 关键活动为 a, d, g, j 。

1. 若在线性表中采用折半查找法查找元素，

该线性表应该（ ）。

- A. 元素按值有序
B. 采用顺序存储结构
C. 元素按值有序，且采用链式存储结构
D. 元素按值有序，且采用顺序存储结构

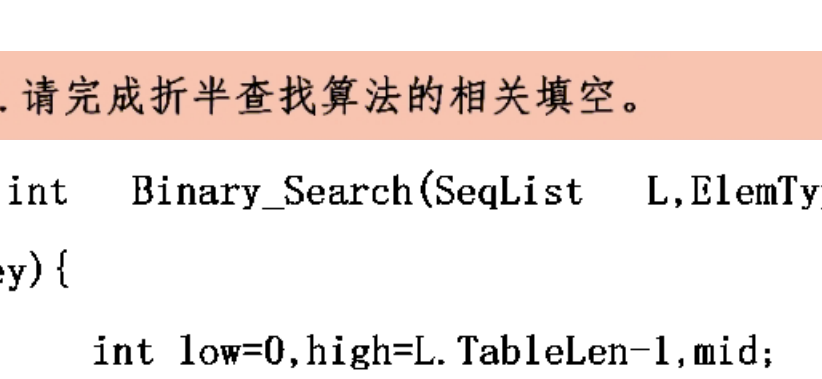
答案：D

2. 折半查找有序表

(4, 6, 10, 12, 20, 30, 50, 70, 88, 100)。若查找表中元素 58，则它将依次与表中（ ）比较大小，查找结果是失败。

- A. 20, 70, 30, 50
B. 30, 88, 70, 50
C. 20, 50
D. 30, 88, 50

答案：A



3. 请完成折半查找算法的相关填空。

```
int Binary_Search(SeqList L, ElemType key) {
    int low=0, high=L.TableLen-1, mid;
    while(_____) {
        mid=_____;
        if(L.elem[mid]==key)
            return mid;
        else if(L.elem[mid]>key)
            _____;
        else
            _____;
    }
}
```

答案：low<=high;

(low+high)/2;

high=mid-1;

low=mid+1

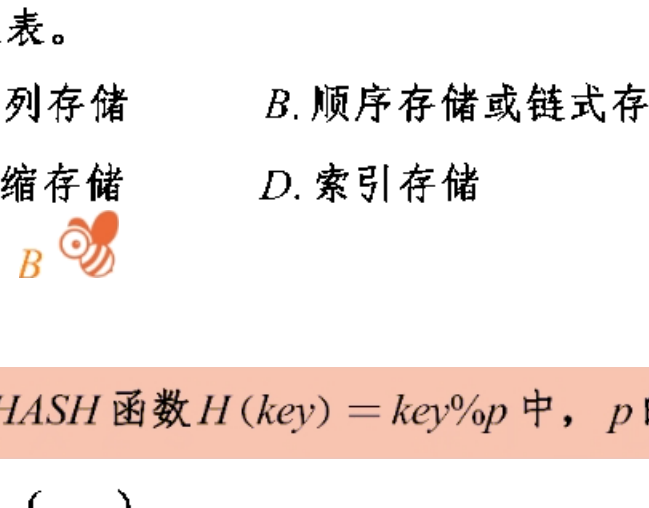
4. 具有 12 个关键字的有序表，折半查找的平

均查找长度为（ ）。

- A. 3.1 B. 4
C. 2.5 D. 5

答案：A

假设 12 个关键字为 1~12



$$ASL = \frac{1}{12} \times (1 \times 1 + 2 \times 2 + 3 \times 4 + 4 \times 5) = 3.08333 \approx 3.1$$

5. 顺序查找法适合于存储结构为（ ）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

答案：B

6. 在 HASH 函数 $H(key) = key \% p$ 中， p 的值

应取（ ）。

- A. 最接近该 HASH 表长（设为 m ，下同）
B. 奇数
C. 小于或等于 m 的最大质数
D. 偶数

答案：C

7. 已知散列表长度为 13，散列函数为

 $H(key) = key \% 11$ ，处理冲突的方法为线性探测法，请画出插入关键字

(10, 8, 40, 27, 21, 57, 46, 23, 19, 56) 以后的散列表，并计算查找成功和不成功时的平均查找长度。

答案：

$$H(10) = 10 \% 11 = 10$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字											10		
比较次数											1		

$$H(8) = 8 \% 11 = 8$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字									8		10		
比较次数									1		1		

$$H(40) = 40 \% 11 = 7$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字								40	8		10		
比较次数								1	1		1		

$$H(27) = 27 \% 11 = 5$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字						27		40	8		10		
比较次数						1		1	1		1		

$$H(21) = 21 \% 11 = 10$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字						27		40	8		10	21	
比较次数						1		1	1		1	2	

$$H(57) = 57 \% 11 = 2$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字			57			27		40	8		10	21	
比较次数			1			1		1	1		1	2	

$$H(46) = 46 \% 11 = 2$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字			57	46		27		40	8		10	21	
比较次数			1	2		1		1	1		1	2	

$$H(23) = 23 \% 11 = 1$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字		23	57	46		27		40	8		10	21	
比较次数		1	1	2		1		1	1		1	2	

$$H(19) = 19 \% 11 = 8$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字		23	57	46		27		40	8	19	10	21	
比较次数		1	1	2		1		1	1	2	1	2	

$$H(56) = 56 \% 11 = 1$$

	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字		23	57	46	56	27		40	8	19	10	21	
比较次数		1	1	2	4	1		1	1	2	1	2	

查找成功的平均查找长度为

$$(1 \times 6 + 2 \times 3 + 4 \times 1) / 10 = 1.6$$

查找不成功的平均查找长度为

$$(1 + 6 + 5 + 4 + 3 + 2 + 1 + 6 + 5 + 4 + 3) / 11 = 40$$

8. 对给定的关键字序列

19, 14, 23, 1, 68, 20, 84, 27, 55, 11，给定哈希函数为 $H(key) = key \% 13$ ，使用拉链法解决冲突建立哈希表，并给出查找成功的平均查找长度。答案：
$$H(19) = 19 \% 13 = 6$$

$$H(14) = 14 \% 13 = 1$$

$$H(23) = 23 \% 13 = 10$$

$$H(1) = 1 \% 13 = 1$$

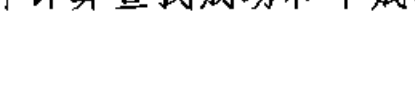
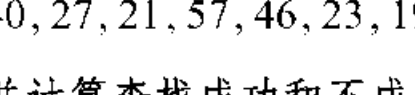
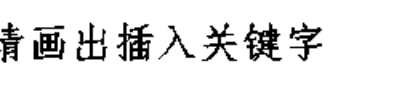
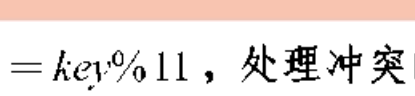
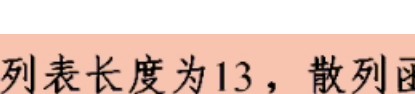
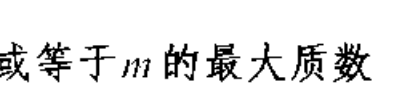
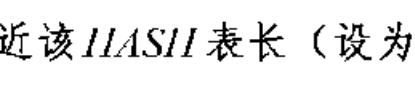
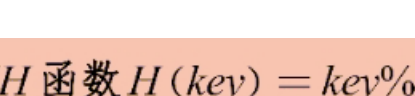
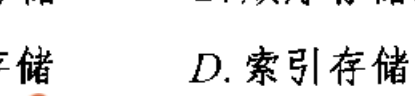
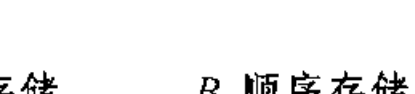
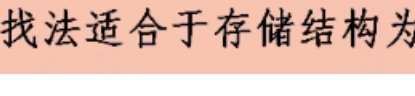
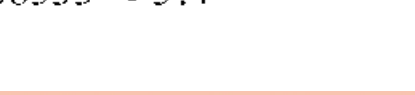
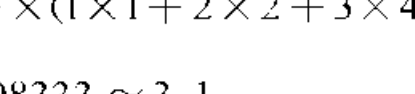
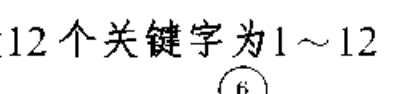
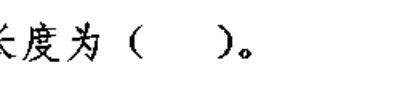
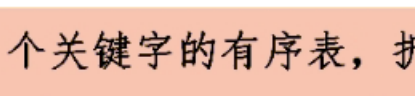
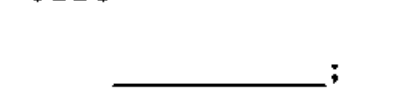
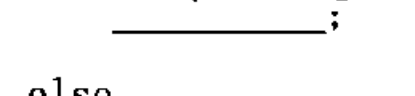
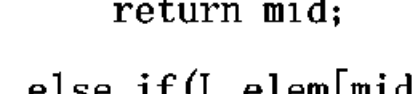
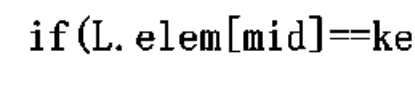
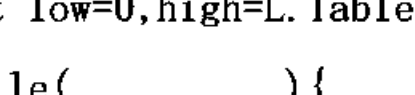
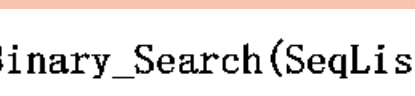
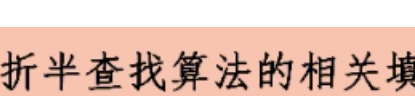
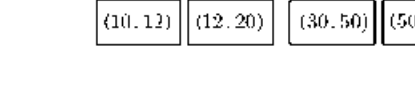
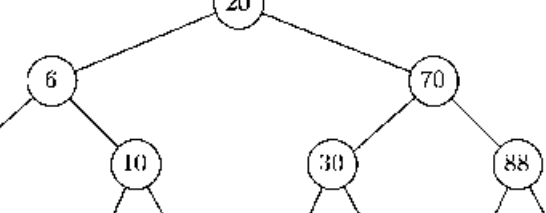
$$H(68) = 68 \% 13 = 3$$

$$H(20) = 20 \% 13 = 7$$

$$H(84) = 84 \% 13 = 6$$

$$H(27) = 27 \% 13 = 1$$

$$H(55) = 55 \% 13 = 3$$

$$H(11) = 11 \% 13 = 11$$



1. 在待排序的记录集中，存在多个具有相同键值的记录，这些记录的相对次序仍然保持不变，称这种排序是稳定排序。（ ）

答案：正确 

2. 若数据元素序列{11, 12, 13, 7, 8, 9, 23, 4, 5}，是采用下列排序方法之一得到的第二趟排序后的结果，则该排序算法只能是（ ）。

A. 冒泡排序 B. 直接插入排序

C. 简单选择排序 D. 二路归并排序

答案：B 


3. 设一组初始记录关键字序列为

{49, 38, 65, 97, 76, 13, 27, 50}，则以 $d = 4$ 为增量的一趟希尔排序结束后的结果为_____。

答案：49, 13, 27, 50, 76, 38, 65, 97 

4. 在对一组记录{18, 6, 27, 12, 52, 15, 47, 29}

进行直接插入排序时，当把第6个记录15插入到有序表时，为寻找插入位置需比较_____次。

答案：4 

与52, 27, 18, 12 比较

5. 直接插入排序算法在平均情况下的时间复

杂度为_____，在最好情况下的时间复杂度为_____。

答案： $O(n^2)$; $O(n)$ 

最好情况下，也就是序列中的所有元素已有序，只需比较而不用移动，此时时间复杂度为 $O(n)$ 。

6. 用直接插入排序对下面四个序列进行递增


排序，元素比较次数最少的是（ ）。

A. 94, 32, 40, 90, 80, 46, 21, 69

B. 32, 40, 21, 46, 69, 94, 90, 80

C. 21, 32, 46, 40, 80, 69, 90, 94

D. 90, 69, 80, 46, 21, 32, 94, 40

答案：C 

C 选项已基本有序。

7. 从未排序序列中选择一个元素，该元素将


未排序序列分成前后两个部分，前一部分中所有元素都小于等于所选元素；后一部分中所有元素都大于等于所选元素，而所选元素处在排序的最终位置，这种排序方法称作（ ）。

A. 插入排序

B. 希尔排序

C. 快速排序

D. 堆排序

答案：C 

8. 已知待排序记录的关键字序列

{77, 71, 52, 22, 15, 30, 3}，用冒泡排序法按从小到大顺序写出每趟排序的结果，直到排序结束。

答案：

71, 52, 22, 15, 30, 3, 77

52, 22, 15, 30, 3, 71, 77

22, 15, 30, 3, 52, 71, 77

15, 22, 3, 30, 52, 71, 77

15, 3, 22, 30, 52, 71, 77

3, 15, 22, 30, 52, 71, 77

或者

3, 77, 71, 52, 22, 15, 30 

3, 15, 77, 71, 52, 22, 30

3, 15, 22, 77, 71, 52, 30

3, 15, 22, 30, 77, 71, 52

3, 15, 22, 30, 52, 77, 71

3, 15, 22, 30, 52, 71, 77

9. 用某种排序方法对关键字序列

{25, 84, 21, 47, 15, 27, 68, 35, 20} 进行排序时，序列的变化情况如下：

20, 15, 21, 25, 47, 27, 68, 35, 84

15, 20, 21, 25, 35, 27, 47, 68, 84

15, 20, 21, 25, 27, 35, 47, 68, 84


则所采用的排序方法是（ ）。

A. 选择排序

B. 希尔排序

C. 归并排序

D. 快速排序

答案：D 

10. 给出一组关键字{29, 18, 25, 47, 58, 12, 51, 10}

写出快速排序方法（第1个元素为枢轴）进行排序的每一趟排序。

答案：

10, 18, 25, 12, 29, 58, 51, 47 

10, 18, 25, 12, 29, 47, 51, 58

10, 12, 18, 25, 29, 47, 51, 58



1. 对具有 n 个元素的任意序列采用选择排序,

排序趟数为 ()。

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

答案: A

2. 堆是完全二叉树, 完全二叉树不一定是堆。

()

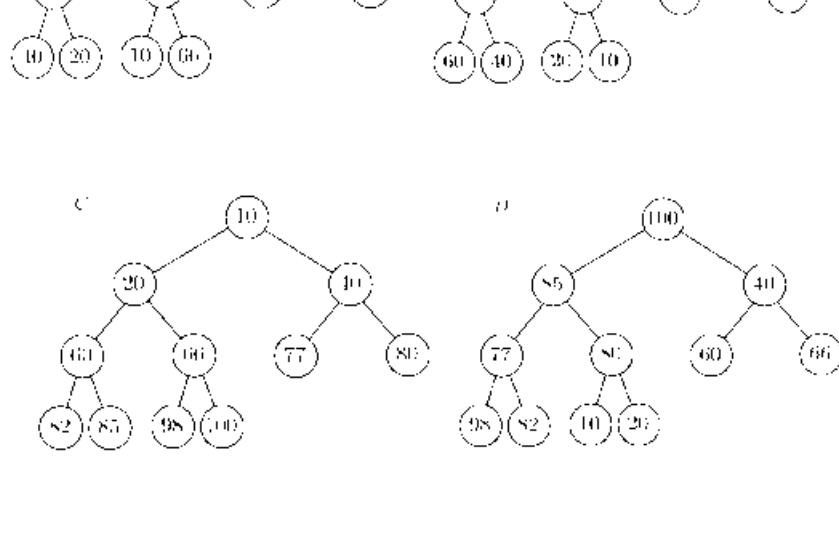
答案: 正确

3. 以下序列不是堆 (大根堆或小根堆) 的是

()。

- A. {100, 85, 98, 77, 80, 60, 82, 40, 20, 10, 66}
B. {100, 98, 85, 82, 80, 77, 66, 40, 20, 10}
C. {10, 20, 40, 60, 66, 77, 80, 82, 85, 98, 100}
D. {100, 85, 40, 77, 80, 60, 66, 98, 82, 10, 20}

答案: D

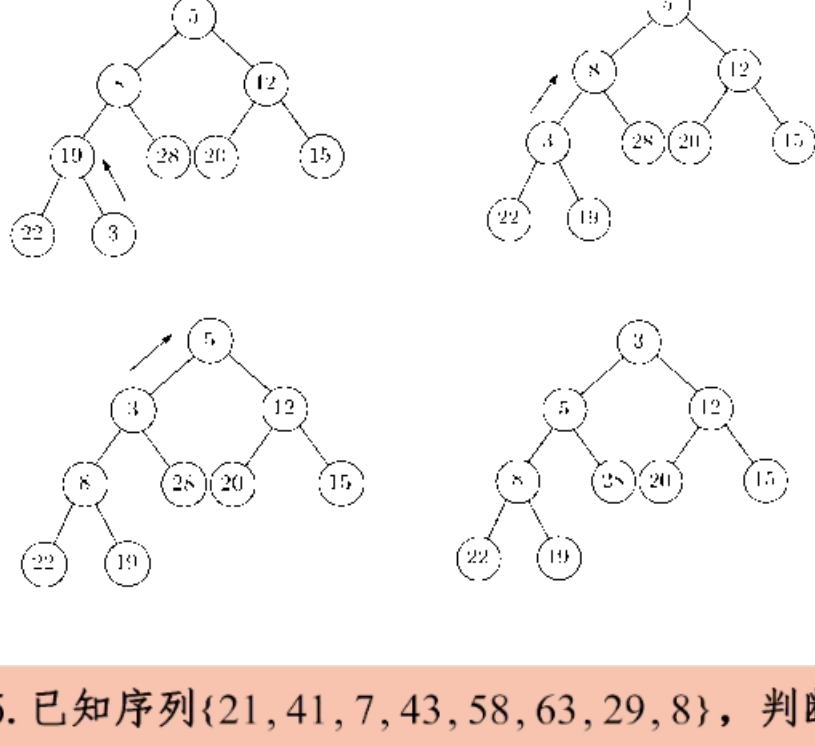


4. 已知关键字序列

{5, 8, 12, 19, 28, 20, 15, 22} 是小根堆, 插入关键字 3, 调整后得到的小根堆是 ()。

- A. 3, 5, 12, 8, 28, 20, 15, 22, 19
B. 3, 5, 12, 19, 20, 15, 22, 8, 28
C. 3, 8, 12, 5, 20, 15, 22, 28, 19
D. 3, 12, 5, 8, 28, 20, 15, 22, 19

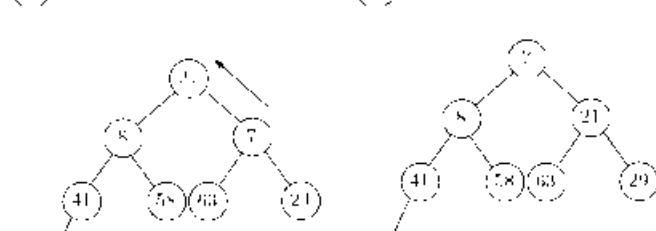
答案: A



5. 已知序列 {21, 41, 7, 43, 58, 63, 29, 8}, 判断

该序列是否为堆。若不是, 将其调整为小根堆, 画出调整完的小根堆。

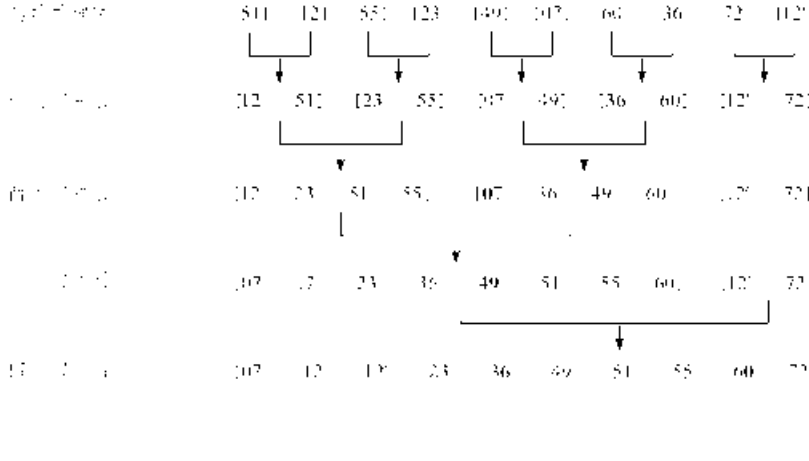
答案: 不是堆



6. 将初始关键字序列

{51, 12, 55, 23, 49, 07, 60, 36, 72, 12}, 写出归并排序算法的每一趟结果。

答案:



7. 给出关键字序列

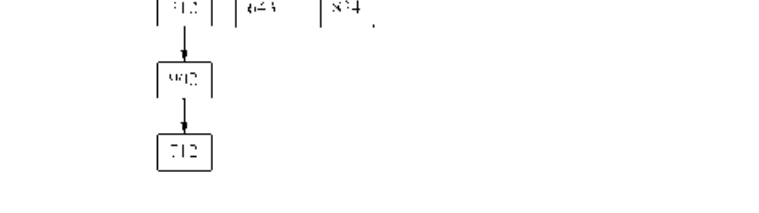
{112, 204, 312, 902, 156, 712, 451, 623, 643, 834}

进行基数排序时每一趟的结果。

答案:

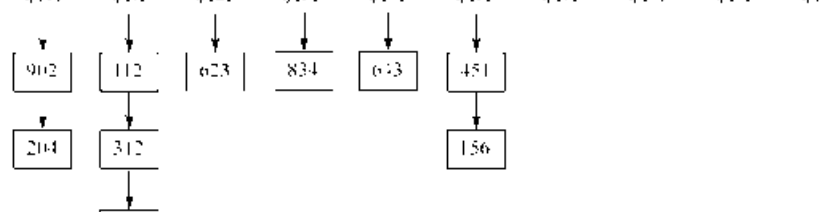
→ [112] → [204] → [312] → [902] → [156] → [712] → [451] → [623] → [643] → [834]

一趟基于个位:



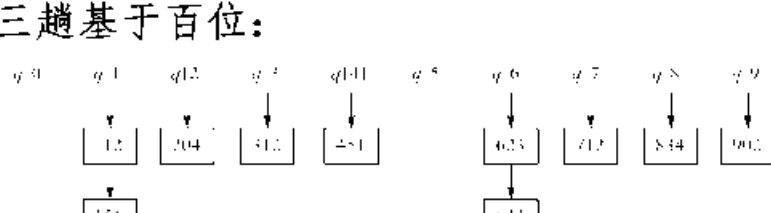
→ [451] → [112] → [312] → [902] → [712] → [623] → [643] → [204] → [834] → [156]

第二趟基于十位:



→ [902] → [204] → [112] → [312] → [712] → [623] → [834] → [643] → [451] → [156]

第三趟基于百位:



→ [112] → [156] → [204] → [312] → [451] → [623] → [643] → [712] → [834] → [902]

8. 在最好和最坏情况下的时间复杂度均为

 $O(n \log_2 n)$ 且稳定的排序方法是 ()。

- A. 快速排序 B. 堆排序
C. 归并排序 D. 基数排序

答案: C

9. 在直接插入排序、希尔排序、冒泡排序和

快速排序中, 平均情况下_____最快, 其时间复杂度为_____。

答案: 快速排序; $O(n \log_2 n)$

10. 下列排序算法中, 第一趟排序结束后, 其

最大或最小元素一定在其最终位置上的算法是 ()。

- A. 归并排序 B. 直接插入排序
C. 快速排序 D. 冒泡排序

答案: D