

26 竞成《数据结构》-选择题部分做题本

第2章绪论

2.5 绪论-习题精编(答案见原书页码 P30)

(1) 某算法的时间复杂度为 $O(n^2)$, 表明该算法的 ()。

- A. 问题规模是 n^2
- B. 执行时间等于 n^2
- C. 执行时间与 n^2 成正比
- D. 问题规模与 n^2 成正比

(2) 下面所示的各个数据结构中,属于非线性数据结构的是()。

A. 树结构

B. 队列结构

C. 栈结构

D. A 、 B 、 C 均是

(3) 下面所示的各个选项中,() 属于逻辑结构。

A. 顺序表

B. 散列表

C. 有序表

D. 单链表

(4) 以下算法的时间复杂度为()。

```
void test(int n){  
    int i=-520;  
    for(i=0; i*i*i<=n;)  
        ++i;  
}
```

A. $O(n)$

B. $O(n \log n)$

C. $O(n^{1/3})$

D. $O(n^{1/2})$

(5) 以下代码在最坏情况下的时间复杂度为()。

```
for(k=n-1; k>=1; --k)
    for(t=1; t<k; ++t)
        if(A[t]>A[t+1])
            swap(A[t], A[t+1])
```

A. $O(n)$

B. $O(n\log n)$

C. $O(n^3)$

D. $O(n^2)$

(6) 下面算法中, 语句 " $x *= 2;$ " 的执行次数是 ()。

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

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

B. $n\log_2 n$

C. n^2

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

(7) 若下列代码中 *combine* 函数的时间复杂度为 $O(n)$ 。则 *combineSort* 函数的时间复杂度是()。

```
void combineSort(int i, int j){  
    if(i != j){  
        int k = (i+j)/2;  
        combineSort(i, k);  
        combineSort(k+1, j);  
        combine(i, j, k); //本函数的时间复杂度为O(n)  
    }  
}
```

A. $O(n^2)$

B. $O(n^3)$

C. $O(n^{3/2})$

D. $O(n\log_2 n)$

(8) 下列各个选项的说法中, 表述有错误的是()。

I. 算法原地工作的含义是指不需要任何额外的辅助空间

II. 在相同规模 n 下, 当 n 足够大时, 复杂度为 $O(n)$ 的算法时间上总优于 $O(2^n)$ 的算法

III. 所谓时间复杂度, 是指最坏情况下估算算法执行时间的一个上界

IV. 同一个算法, 实现语言的级别越高, 执行效率越低

A. *I*

B. *I, II*

C. *I, IV*

D. *III*

(9) 计算机算法指的是 (①), 它必须具备 (②) 这三个特性。

(1) *A.*计算方法 *B.*排序方法 *C.*解决问题的步骤序列 *D.*调度方法

(2) *A.*可执行性、可移植性、可扩充性 *B.*可执行性、确定性、有穷性

*C.*确定性、有穷性、稳定性 *D.*易读性、稳定性、安全性

(10) 下列关于数据结构的说法中错误的是()。

- A. 数据结构相同, 对应的存储结构也相同
- B. 数据结构涉及数据的逻辑结构、存储结构和施加在其上的操作
- C. 数据结构操作的实现与存储结构有关
- D. 定义逻辑结构时可以不考虑存储结构

(11) 下列说法中, 不正确的是()。

- A. 数据元素是数据的基本单位
- B. 数据项是数据元素中不可分割的最小可标识单位
- C. 数据可由若干数据元素构成
- D. 数据项可由若干个数据元素构成

(12) 数据的四种基本存储结构是指 ()。

- A. 顺序存储结构、索引存储结构、直接存储结构、倒排存储结构
- B. 顺序存储结构、索引存储结构、链式存储结构、散列存储结构
- C. 顺序存储结构、非顺序存储结构、指针存储结构、树型存储结构
- D. 顺序存储结构、链式存储结构、树型存储结构、图型存储结构

(13) 下面关于“算法”的描述, 错误的是()。

- A. 算法必须是正确的
- B. 算法必须要能够结束
- C. 一个问题可以有多种算法解决
- D. 算法的某些步骤可以有二义性

(14) 某算法的空间复杂度为 $O(1)$, 则 ()。

- A. 该算法执行不需要任何辅助空间
- B. 该算法执行所需辅助空间大小与问题规模 n 无关
- C. 该算法执行不需要任何空间
- D. 该算法执行所需空间大小与问题规模 n 无关

2.6 绪论-真题演练(答案见原书页码 P30)

(18) 【2011】 设 n 是描述问题规模的非负整数, 下面程序片段的时间复杂度是 ()。

```
x=2;  
while(x<n/2);  
x=2*x;
```

A. $O(\log_2 n)$

B. $O(n)$

C. $O(n \log_2 n)$

D. $O(n^2)$

(19) 【2012】求整数 $n(n \geq 0)$ 阶乘的算法如下, 其时间复杂度是 ()。

```
int fact(int n){  
    if(n<=1) return 1;  
    return n*fact(n-1);  
}
```

A. $O(\log_2 n)$

B. $O(n)$

C. $O(n \log_2 n)$

D. $O(n^2)$

(20) 【2014】下列程序段的时间复杂度是()。

```
count=0;
for(k=1; k<=n; k*=2)
    for(j=1; j<=n; j++)
        count++
```

A. $O(\log_2 n)$

B. $O(n)$

C. $O(n \log_2 n)$

D. $O(n^2)$

(21) 【2017】下列函数的时间复杂度是()。

```
int func(int n){  
    int i=0,sum=0;  
    while(sum<n) sum+=++i;  
    return i;  
}
```

A. $O(\log_2 n)$

B. $O(n^{1/2})$

C. $O(n)$

D. $O(n\log_2 n)$

(22) 【2019】设 n 是描述问题规模的非负整数, 下列程序段的时间复杂度是 ()。

```
x=0;  
while(n>=(x+1)(x+1))  
    x=x+1
```

A. $O(\log_2 n)$

B. $O(n^{1/2})$

C. $O(n)$

D. $O(n^2)$

第3章线性表

3.1.4 线性表的基本概念-习题精编(答案见原书页码 P37)

(1) 在线性表中, 除开始元素外, 每个元素(), 除结束元素外, 每个元素()。

- A. 只有唯一的前驱 B. 只有唯一的后继 C. 不止一个前驱 D. 不止一个后继

(2) 线性表是一个 ()。

A. 有限序列,可以为空

B. 有限序列,不能为空

C. 无限序列,可以为空

D. 无限序列,不能为空

(3) 线性表是具有 n 个 () 的有限序列。

A. 表元素

B. 数据元素

C. 数据项

D. 信息项

(4) 以下关于链式存储结构的叙述中,正确的是()。

- A. 存储密度大于顺序存储结构
- B. 逻辑上相邻的结点物理上不必相邻
- C. 支持随机访问
- D. 插入删除不方便, 需要移动结点

3.2.3 线性表的顺序存储-习题精编(答案见原书页码 P44)

(1) 线性表的顺序存储结构是一种 () 的存储结构。

- A. 随机存取 B. 散列存取 C. 索引存取 D. 顺序存取

(2)(多选) 一个顺序表占用存储空间的大小与 () 有关。

- A. 表容量 *capacity* B. 单个元素的长度 C. 表中元素数量 D. 元素的物理位置

(3) 在 n 个结点的线性表的数组实现中, 算法的时间复杂度是 $O(1)$ 的操作是 ()。

- A. 访问第 $i(1 \leq i \leq n)$ 个结点和求第 $i(2 \leq i \leq n)$ 个结点的直接前驱
- B. 在第 $i(1 \leq i \leq n)$ 个结点后插入一个新结点
- C. 删除第 $i(1 \leq i \leq n)$ 个结点
- D. 以上都不对

(4) 采用动态分配存储空间的顺序表, 若已存放 n 个元素并存满存储空间, 需要再存放 m 个元素, 此时申请空间成功是说明线性表至少有 () 个地址连续可分配的空闲空间。

A. m B. $m - n$ C. $n + m$ D. n

(5) 对于长度为 n 的顺序表, 删除其中任何一个位置元素的概率相等。则删除一个元素时, 平均需要移动 () 个元素。

A. $(n+1)/2$

B. $n/2$

C. $(n-1)/2$

D. $(n-2)/2$

(6) 向一个有 n 个元素的顺序表中插入一个新元素, 平均要移动 () 个元素。

A. 1

B. 0

C. $n/2$

D. $(n-1)/2$

3.3.7 线性表的链式存储-习题精编(答案见原书页码 P66)

(1) 在一个单链表中, 若 p 所指的结点不是最后一个结点, 在 p 之后插入 s 所指的结点, 则需要执行 ()。

A. $s \rightarrow next = p; p \rightarrow next = s;$

B. $p \rightarrow next = s; s \rightarrow next = p;$

C. $p = s; s \rightarrow next = p \rightarrow next;$

D. $s \rightarrow next = p \rightarrow next; p \rightarrow next = s;$

(2) 将长度为 n 的单向链表链接在长度为 m 的单向链表之后的算法时间复杂度为 ()。

A. $O(1)$

B. $O(n)$

C. $O(m)$

D. $O(m+n)$

(3) 单链表中, 增加一个头结点的目的是 ()。

- A. 使单链表至少有一个结点
- B. 标识表中首结点的位置
- C. 方便运算的实现
- D. 说明单链表是线性表的链式存储

(4) 在一个有尾指针的无头结点单链表上, 则下列与表长有关的操作是 ()。

- A. 删除头指针所指的元素
- B. 删除尾指针所指的元素
- C. 在单链表第一个元素前插入一个元素
- D. 在单链表最后一个元素后插入一个元素

(5) 将一个元素插入长度为 n 的有序单链表中, 如果插入元素之后, 单链表仍保持有序, 则该插入操作的时间复杂度为 ()。

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

(6) 将两个长度为 n 的有序链表归并成一个有序表, 最少比较 () 次。

A. n

B. $2n - 1$

C. $2n$

D. $n - 1$

(7) 在一个单链表中, 若 p 所指的结点不是最后一个结点, 删除 p 之后结点, 则执行 ()。

A. $p \rightarrow next = p;$

B. $p \rightarrow next \rightarrow next = p \rightarrow next;$

C. $p \rightarrow next \rightarrow next = p;$

D. $p \rightarrow next = p \rightarrow next \rightarrow next;$

(8) h 为不带头结点的单链表。在 h 的表头上插入一个新结点 t 的语句是 ()。

A. $t \rightarrow next = h; h = t;$

B. $h = t; t \rightarrow next = h;$

C. $t \rightarrow next = h \rightarrow next; h = t;$

D. $h = t; t \rightarrow next = h \rightarrow next;$

(9) 在一个单链表中, 已知 q 指向结点是 p 指向结点的前驱结点, 若在 q 指向结点和 p 指向结点之间插入 s 指向结点, 则需执行 ()。

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

(10) 在一个单链表中, 已知指针 p 指向其中某个结点, 若在该结点前插入一个由指针 s 指向的结点, 则需执行 ()。

- A. $s \rightarrow next = p \rightarrow next; p \rightarrow next = s;$
- B. $p \rightarrow next = s; s \rightarrow next = p;$
- C. $r = p \rightarrow next; p \rightarrow next = s; s \rightarrow next = r;$
- D. 仅靠已知条件无法实现

(11) 对于没有尾指针的单链表, 将 n 个元素采用头插法建立单链表的时间复杂度为 (), 采用尾插法建立单链表的时间复杂度为 ()。

A. $O(1)$

B. $O(n \log n)$

C. $O(n)$

D. $O(n^2)$

(12) 双链表相比于单链表 ()。

- A. 插入删除操作更简单
- B. 不需要尾指针
- C. 访问邻结点更灵活
- D. 支持随机访问

(13) 在双链表中, 将 s 指向的结点插入到 p 结点之前, 需要完成操作为 ()。

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

(14) 在双链表中, 删除 p 指针所指结点的操作应为()。

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

(15) 在一个双链表中, 在 p 指针所指的结点之后插入 s 指针所指结点的操作应为 ()。

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

(16) 带头结点的双循环链表 L 如果为空, 则需要满足的条件为 ()。

- A. $L \rightarrow prev == L$ and $L \rightarrow next == NULL$
- B. $L \rightarrow prev == L$ and $L \rightarrow next == L$
- C. $L \rightarrow prev == NULL$ and $L \rightarrow next == NULL$
- D. $L \rightarrow prev == NULL$ and $L \rightarrow next == L$

(17) (多选) 某线性表用带头结点的循环单链表存储, 头指针为 $head$, 当 $head \rightarrow next \rightarrow next \rightarrow next == head$ 成立时, 线性表长度可能是 ()。

A. 0

B. 1

C. 2

D. 3

(18) 设单循环链表中结点的结构为 $(data, next)$, 且 $rear$ 是指向非空的带头结点的单循环链表的尾结点的指针。若要删除链表的第一个结点, 正确的操作是 ()。

- A. $s = rear; rear = rear \rightarrow next; free(s);$
- B. $rear = rear \rightarrow next; free(s);$
- C. $rear = rear \rightarrow next \rightarrow next; free(s);$
- D. $s = rear \rightarrow next \rightarrow next; rear \rightarrow next \rightarrow next = s \rightarrow next; free(s);$

(19) 若某种用链表实现的数据结构只在第一个元素前插入元素和删除最后一个元素,以下哪种链表最适合实现该数据结构()。

- A. 只带队首指针的非循环双链表
- B. 只带队首指针的循环双链表
- C. 只带队尾指针的循环双链表
- D. 只带队尾指针的循环单链表

(20) 若一个带头结点的链表经常在末尾进行插入删除操作,要求尽量不引入额外变量,则应该选用 ()

- A. 不带尾指针的双循环链表
- B. 不带尾指针的单循环链表
- C. 带尾指针的单循环链表
- D. 带尾指针的双循环链表

(21) 设 p 指向非空单循环链表 (带头结点) 的终端结点, 则 p 满足 ()。

A. $p \rightarrow next == L$

B. $p \rightarrow next = NULL$

C. $p \rightarrow next \rightarrow next == L$

D. $p \rightarrow next = L$ and $p \neq L$

(22) (多选) 以下选项正确的是 ()。

- A. 静态链表既有顺序存储的优点,又有动态链表的优点,所以,它存取表中第 i 个元素的时间与 i 无关
- B. 静态链表中能容纳的元素个数在表定义时就确定了,在后续操作中不能增加
- C. 静态链表与动态链表在元素的插入、删除上类似,不需要做元素的移动
- D. 静态链表相比于动态链表有可能浪费较多存储空间

(23) 设线性表中有 n 个元素, () 选择在单链表上执行相比顺序表更好。

- A. 删除所有值为 e 的元素
- B. 在最后一个元素后面插入新元素 e
- C. 顺序输出线性表
- D. 交换第 i 个元素和第 j 个元素的值

(24) 下面关于线性表的叙述中, 不正确的是()。

*I.*线性表在链式存储时, 查找第 i 个元素的时间同 i 的值成正比

*II.*线性表在链式存储时, 查找第 i 个元素的时间同 i 的值无关

*III.*线性表在顺序存储时, 查找第 i 个元素的时间同 i 的值成正比

*IV.*线性表在顺序存储时, 查找第 i 个元素的时间同 i 的值无关

A. *I*、*II*

B. *II*、*III*

C. *III*、*IV*

D. *I*、*IV*

3.3.8 线性表的链式存储-真题演练(答案见原书页码 P66)

(26) 【2013】已知两个长度分别为 m 和 n 的升序链表, 若将它们合并为一个长度为 $m+n$ 的降序链表, 则最坏情况下的时间复杂度是 ()。

- A. $O(n)$ B. $O(mn)$ C. $O(\min(m,n))$ D. $O(\max(m,n))$

(27) 【2016】已知表头元素为 c 的单链表在内存中的存储状态如下表所示。

地址	元素	链接地址
1000H	a	1010H
1004H	b	100CH
1008H	c	1000H
100CH	d	null
1010H	e	1004H
1014H		

现将 f 存放于 1014H 处并插入到单链表中, 若 f 在逻辑上位于 a 和 e 之间, 则 a, e, f 的“链接地址”依次是 ()。

A. 1010H, 1014H, 1004H

B. 1010H, 1004H, 1014H

C. 1014H, 1010H, 1004H

D. 1014H, 1004H, 1010H

(28) 已知一个带有表头结点的双向循环链表 L , 结点结构为: $prev \text{ data } next$, 其中, $prev$ 和 $next$ 分别是指向其直接前驱和直接后继结点的指针。现要删除指针 p 所指的结点, 正确的语句序列是 ()。

- A. $p \rightarrow next \rightarrow prev = p \rightarrow prev; p \rightarrow prev \rightarrow next = p \rightarrow prev; free(p)$
- B. $p \rightarrow next \rightarrow prev = p \rightarrow next; p \rightarrow prev \rightarrow next = p \rightarrow next; free(p)$
- C. $p \rightarrow next \rightarrow prev = p \rightarrow next; p \rightarrow prev \rightarrow next = p \rightarrow prev; free(p)$
- D. $p \rightarrow next \rightarrow prev = p \rightarrow prev; p \rightarrow prev \rightarrow next = p \rightarrow next; free(p)$

(29) 【2021】 已知头指针 h 指向一个带头结点的非空单循环链表, 结点结构为: $4atanext$ 。其中 $next$ 是指向直接后继结点的指针, p 是尾指针, q 是临时指针。现要删除该链表的第一个元素, 正确的语句序列是 ()。

- A. $h \rightarrow next = h \rightarrow next \rightarrow next; q = h \rightarrow next; free(q);$
- B. $q = h \rightarrow next; h \rightarrow next = h \rightarrow next; free(q);$
- C. $q = h \rightarrow next; h \rightarrow next = q \rightarrow next; if (p \neq q) p = h; free(q);$
- D. $q = h \rightarrow next; h \rightarrow next = q \rightarrow next; if (p == q) p = h; free(q);$

第 4 章 栈、队列和数组

4.1.6 栈-习题精编(答案见原书页码 P81)

(1) 有一个空栈, 栈顶指针为 $3000H$, 每个元素需要一个存储单元, 执行 *Push*、*Push*、*Push*、*Push*、*Pop*、*Push*、*Pop*、*Push* 操作后, 栈顶指针的值为 ()。

A. $3002H$

B. $3003H$

C. $3004H$

D. $3005H$

(2) 设 a, b, c, d, e, f 按照顺序进栈, 允许随时出栈, 则不可能的出栈操作是()。

A. $bcafed$

B. $fedcba$

C. $cbadef$

D. $cadbef$

(3) 用 I 表示进栈, O 表示出栈, 则 $abcd$ 顺序进栈, 允许随时出栈, 得到出栈序列 $bcad$ 的相应的操作是 ()。

A. $IIOIOOIO$

B. $IOIOHIOO$

C. $IIIOOIOO$

D. $IOHOOIO$

(4) 设栈的输入序列是 1,2,3,4, 则不可能是其出栈序列的是()。

A. 1243

B. 2134

C. 1432

D. 4312

(5) 设 n 个元素的进栈序列是 $1, 2, 3, \dots, n$, 其输出序列是 $p_1, p_2, p_3, \dots, p_n$, 若 $p_1 = 3$, 则 p_2 的值()。

A. 一定是 2

B. 一定是 1

C. 不可能是 1

D. 以上都不对

(6) (多选) 若一个栈的入栈序列为 $1, 2, 3, 4$, 其出栈序列为 p_1, p_2, p_3, p_4 , 则 p_2 和 p_4 可能为 ()。

A. 2, 4

B. 2, 1

C. 4, 3

D. 3, 4

(7)(多选)共享栈的好处是()。

- A. 减少存取时间 B. 节省存储空间 C. 降低上溢的可能 D. 降低下溢的可能

(8) 一个栈的输入序列为 $1, 2, 3, \dots, n$, 若输出序列的第一个元素是 n , 输出的第 i ($1 \leq i \leq n$) 个元素是 ()。

A. 不确定

B. $n - i$

C. i

D. $n - i + 1$

(9) 若进栈序列为 1,2,3,4,5, 可能得到的出栈序列是()。

A. 12534

B. 31254

C. 32541

D. 14235

(10) 对于空栈时 $top == 0$ 的栈,进栈操作的操作顺序是(),出栈操作的操作顺序是()。

①判断栈是否为空栈；②判断栈是否为满栈；③ top 自增；④ top 自减；⑤插入或删除

A. ①⑤④

B. ②③⑤

C. ②⑤③

D. ①④⑤

(11) 一个栈的输入序列为 $1, 2, 3, \dots, n$, 若存在 $k > 1$ 使第 k 个输出的元素为 n , 则输出的第 i ($i > k$) 个元素是 ()。

A. 不确定

B. $n - i - k$

C. $n - i + k$

D. $n - i + 1$

(12) 有六个元素 6,5,4,3,2,1 的顺序进栈, 则下列不是合法出栈序列的是 ()。

A. 543612

B. 453126

C. 346521

D. 234156

(13) C 语言标识符的第一个字符不能是数字, 当字符序列“t3_”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. 4

B. 5

C. 6

D. 3

(14) 若栈采用顺序存储方式存储, 现两栈共享空间 $V[1 \cdots m]$, $top[i]$ 代表第 i 个栈栈顶, 栈 1 的底在 $V[1]$, 栈 2 的底在 $V[m]$, 则栈满的条件是 ()。

A. $|top[2] - top[1]| == 0$

B. $top[1] + 1 = top[2]$

C. $top[1] + top[2] == m$

D. $top[1] == top[2]$

(15) 对于单词 *bookkeeper*, 按字母顺序进栈, 则有多少种出栈顺序使其依然是原单词 ()。

A. 8

B. 20

C. 12

D. 16

(16) 设栈的容量为2, 以1,2,3,4,5的顺序入栈(可随时出栈), 出栈序列可能是()。

A. 12345

B. 54321

C. 32145

D. 21543

4.1.7 栈-真题演练(答案见原书页码 P81)

(18) 【2010】若元素 a, b, c, d, e, f 依次进栈, 允许进栈、退栈操作交替进行, 但不允许连续三次进行退栈操作, 则不可能得到的出栈序列是 ()。

A. $dcebfa$ B. $cbdaef$ C. $bcaefd$ D. $afedcb$

(19) 【2011】元素 a, b, c, d, e 依次进入初始为空的栈中, 若元素进栈后可停留、可出栈, 直到所有元素都出栈, 则在所有可能的出栈序列中, 以元素 d 开头的序列个数是 ()。

A. 3

B. 4

C. 5

D. 6

(20) 【2013】一个栈的入栈序列为 $1, 2, 3, \dots, n$, 其出栈序列是 $p_1, p_2, p_3, \dots, p_n$ 。若 $p_2 = 3$, 则 p_3 可能取值的个数是()。

A. $n - 3$

B. $n - 2$

C. $n - 1$

D. 无法确定

(21) 【2018】若栈 S_1 中保存整数, 栈 S_2 中保存运算符, 函数 $F()$ 依次执行下述各步操作:

- ①从 S_1 中依次弹出两个操作数 a 和 b ;
- ②从 S_2 中弹出一个运算符 op ;
- ③执行相应操作 $bopa$;
- ④将运算结果压入 S_1 中。

假定 S_1 中的操作数依次是 5, 8, 3, 2 (2 在栈顶), S_2 中的运算符依次是 * - + (+ 在栈顶)。调用 3 次 $F()$ 后, S_1 栈顶保存的值是 ()。

- A. - 15
- B. 15
- C. - 20
- D. 20

(22) 【2020】对空栈 S 进行 $Push$ 和 Pop 操作, 入栈序列 a, b, c, d, e , 经过 $Push, Push, Pop, Push, Pop, Push, Push, Pop$ 操作后, 得到的出栈序列是 ()。

A. b, a, c

B. b, a, e

C. b, c, a

D. b, c, e

4.2.7 队列-习题精编(答案见原书页码 P90)

(1) 栈的特点是 (1), 队列的特点是 (2), 栈和队列都是 (3)。若进栈序列为 1234, 则 (4) 不可能是一个出栈序列 (不一定全部进栈后再出栈); 若进队列的序列为 1234, 则 (5) 是一个出队列序列。

(1) A. 先进先出 B. 后进先出 C. 进优于出 D. 出优于进

(2) A. 先进先出 B. 后进先出 C. 进优于出 D. 出优于进

(3) A. 顺序存储的线性结构 B. 链式存储的线性结构
C. 限制存取点的线性结构 D. 限制存取点的非线性结构

(4) A. 4231 B. 1324 C. 3241 D. 3214

(5) A. 4231 B. 1234 C. 3241 D. 3214

(3) 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 和 e_6 依次通过栈 S , 一个元素出栈后即进队列 Q , 若 6 个元素的出队序列是 e_2 、 e_4 、 e_3 、 e_6 、 e_5 、 e_1 , 则栈 S 的容量至少应该是 ()。

A. 6

B. 4

C. 3

D. 2

(4) 已知循环队列存储在 C 数组 $A[m]$ 中, $front$ 指向队头元素的前一个位置, $rear$ 指向队尾元素, 则当前队列中的元素个数为 ()。

A. $(rear - front + m) \% m$

B. $(rear - front - m + 1) \% m$

C. $(rear - front) \% m$

D. $(rear - front - 1) \% m$

(5) 若用一个大小为 6 的数组来实现循环队列,且当前 *rear* 和 *front* 的值分别为 0 和 3,当从队列中删除一个元素,再加入两个元素后,*rear* 和 *front* 的值分别为()。

A. 1、5

B. 2、4

C. 4、2

D. 5、1

(6) 假设一个循环队列 Q 仅由队头指针 $front$, 队尾指针 $rear$ 和最大容量 m 组成, 则判断该队列队满的条件是 ()。

A. $front == (rear + 1) \% m$

B. $front == rear + 1$

C. $front == rear$

D. $rear == (front + 1) \% m$

(7) 链队列由非循环单链表表示, 队头是表尾, 不设表尾指针, 进队操作的时间复杂度是()。

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

(8) 循环队列与顺序队列的容量都为 $MaxSize$, 其头指针为 $front$, 尾指针为 $rear$, 则空队列的条件分别为 ()。

A. $(rear + 1) \% MaxSize == front$

B. $rear == front$

C. $(front + 1) \% MaxSize == rear$

D. $rear == 0$

(9) 设循环队列中数组的下标为 $0 \sim N-1$, 已知其队头指针 f (f 指向队首元素的前一个位置) 和队中元素个数 n , 则队尾指针 r (r 指向队尾元素的位置) 为 ()。

A. $f - n$

B. $(f - n) \% N$

C. $(f + n) \% N$

D. $(f + n + 1) \% N$

(10) 最不适合用作链式队列的链表是()。

- A. 只带队首指针的非循环双链表
- B. 只带队首指针的循环双链表
- C. 只带队尾指针的循环双链表
- D. 只带队尾指针的循环单链表

(11) 已知循环队列存储在数组 $A[m]$ 中, $length$ 表示循环队列中的元素个数, $rear$ 指向队尾元素, 插入元素时执行 $rear = (rear + 1) \% m$, 则 $front$ 所在位置应该是 ()。

A. $rear - length$

B. $(rear - length + m) \% m$

C. $(rear + m + 1 - length) \% m$

D. $(rear + length - 1) \% m$

(12) 在链队列中, 队首指针为 $front$, 队尾指针为 $rear$, 将 x 所指向的结点入队, 其操作为 ()。

- A. $front = x; front = front \rightarrow next;$
- B. $x \rightarrow next = front \rightarrow next; front = x;$
- C. $rear \rightarrow next = x; rear = x;$
- D. $rear \rightarrow next = x; x \rightarrow next = null; rear = x;$

(13) (多选) 已知输入序列为 $abcd$, 经过输出受限的双向队列后能得到的输出序列有 ()。

A. $dacb$

B. $cabd$

C. $dbca$

D. $bdac$

(14) 设输入元素序列为 1,2,3,4,5, 利用两个队列, 下面哪种排列不可能得到? ()。

A. 1,2,3,4,5

B. 5,2,3,4,1

C. 1,3,2,4,5

D. 4,1,5,2,3

4.2.8 队列-真题演练(答案见原书页码 P90)

(15) 【2009】设栈 S 和队列 Q 的初始状态均为空, 元素 a 、 b 、 c 、 d 、 e 、 f 、 g 依次进入栈 S 。若每个元素出栈后立即进入队列 Q , 且 7 个元素出队的顺序是 b 、 d 、 c 、 f 、 e 、 a 、 g , 则栈 S 的容量至少是 ()。

A. 1

B. 2

C. 3

D. 4

(16) 【2010】某队列允许在其两端进行入队操作, 但仅允许在一端进行出队操作。若元素 a 、 b 、 c 、 d 、 e 依次入此队列后再进行出队操作, 则不可能得到的出队序列是 ()。

A. $bacde$

B. $dbace$

C. $dbcae$

D. $ecbad$

(17) 【2011】 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。若初始时队列为空, 且要求第 1 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. $n-1, n-1$

(18) 【2014】循环队列放在一维数组 $A[0, \dots, M-1]$ 中, $end1$ 指向队头元素, $end2$ 指向队尾元素的后一个位置。假设队列两端均可进行入队和出队操作, 队列中最多能容纳 $M-1$ 个元素。初始时为空。下列判断队空和队满的条件中, 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. 队空: $end2 == (end1 + 1) \bmod M$; 队满: $end1 == (end2 + 1) \bmod M$;
- D. 队空: $end1 == (end2 + 1) \bmod M$; 队满: $end2 == (end1 + 1) \bmod (M - 1)$;

(19) 【2018】现有队列 Q 与栈 S , 初始时 Q 中的元素依次是 1、2、3、4、5、6(1 在队头), S 为空。若仅允许下列三种操作: ①出队并输出出队元素; ②出队并将出队元素入栈; ③出栈并输出出栈元素, 则不能得到的输出序列是 ()。

A. 1、2、5、6、4、3

B. 2、3、4、5、6、1

C. 3、4、5、6、1、2

D. 6、5、4、3、2、1

(20) 【2021】已知初始为空的队列 Q 的一端仅能进行入队操作, 另外一端既能进行入队操作又能进行出队操作, 若 Q 的入队序列是 1、2、3、4、5 则不能得到的出队序列是()。

A. 5、4、3、1、2

B. 5、3、1、2、4

C. 4、2、1、3、5

D. 4、1、3、2、5

4.3.3 栈与队列的应用-习题精编(答案见原书页码 P99)

(1)(多选)栈可以应用在()。

- A. 递归、表达式求解
- B. 表达式转化、缓冲区
- C. 迷宫求解、括号匹配
- D. 进制转换

(2)(多选) 队列可以运用在()。

A. 缓冲区

B. 迷宫求解

C. 递归

D. 消息队列

(3) 下列说法中, 正确的是()。

- A. 消除递归必须用栈
- B. 函数调用通常会使用队列
- C. 队列和栈都允许在两端进行操作
- D. 表达式求值可以不借助栈

(4) 通常情况下, 一个问题的递归算法相比于非递归算法而言 ()。

A. 效率更高

B. 效率更低

C. 两者相同

D. 无法比较

(5) 中缀表达式 $(A + B) * (C - D) / (E - F * G)$ 的后缀表达式是 ()。

A. $A + B * C - D / E - F * G$

B. $AB + CD - * EFG * - /$

C. $AB + C * D - E / F - G *$

D. $ABCDEF * + * - / - *$

(6) 利用栈求表达式的值时, 设立操作符栈 SK , 若栈 SK 只有两个存储单元, 在下列表式中, 不发生上溢的是 ()。

A. $A - B * (C - D)$

B. $(A - B) * C - D$

C. $(A - B * C) - D$

D. $(A - B) * (C - D)$

(7)C语言函数 *calc* 的定义如下, *a*, *b* 均为正整数, 则 *calc(a, b)* 的运行结果是 ()。

```
int calc(int x, int y){  
    if(y==1) return x;  
    else return calc(x, y-1)+x;  
}
```

A. $a * (b-1)$

B. $a * b$

C. $a + b$

D. $a + a$

(8) 设递归函数如下,问 $func(func(5))$ 在执行过程中,第4个开始被执行的 $func$ 是 ()。

```
int func(int x){  
    if(x<=3) return 2;  
    else return func(x-2)+func(x-4);  
}
```

A. $func(2)$

B. $func(3)$

C. $func(4)$

D. $func(5)$

(9) 用栈来检查算数表达式中的括号是否匹配的方法是: 初始栈为空, 从左到右扫描表达式, 遇到“(”就将其入栈, 遇到“)”就执行出栈操作。检查算术表达式“ $((a + b/(a + b) - c/a)/b)$ ”时, 由于 (), 该表达式中的括号不匹配。

- A. 栈为空却要进行出栈操作
- B. 栈已满却要进行入栈操作
- C. 表达式处理已结束, 栈中仍留有字符“(”
- D. 表达式处理已结束, 栈中仍留有字符“)”

4.3.4 栈与队列的应用-真题演练(答案见原书页码 P99)

(10) 【2009】为解决计算机主机与打印机之间速度不匹配问题,通常设置一个打印数据缓冲区,主机将要输出的数据依次写入该缓冲区,而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是()。

A. 栈

B. 队列

C. 树

D. 图

(11) 【2012】已知操作符包括 $+$ 、 $-$ 、 $*$ 、 $/$ 、 $($ 、 $)$ 。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 $ab + acd + e/f - * - g +$ 时,用栈来存放暂时还不能确定运算次序的操作符,若栈初始时空,则转换过程中同时保存在栈中的操作符的最大个数是()。

A. 5

B. 7

C. 8

D. 11

(12) 【2014】假设栈初始为空, 将中缀表达式 $a/b + (c*d - e*f)/g$ 转换为等价的后缀表达式的过程中, 当扫描到 f 时, 栈中的元素依次是 ()。

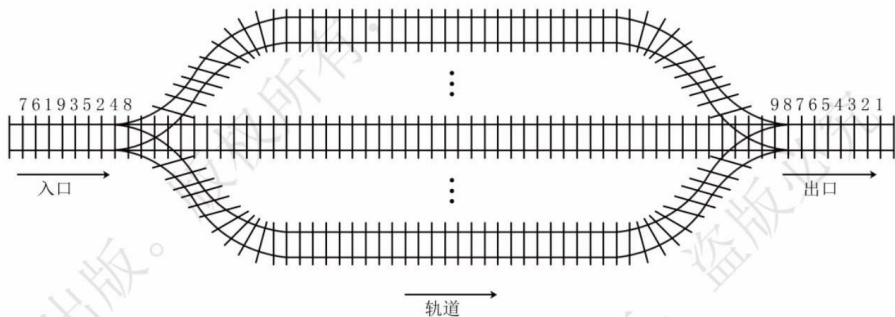
- A. $+ (* -$ B. $+ (- *$ C. $/ + (* - *$ D. $/ + - *$

(13) 【2015】已知程序如下, 程序运行时使用栈来保存调用过程的信息, 自栈底到栈顶保存的信息依次对应的是()。

```
int S(int n) {return(n<=0) ?0 :S(n-1)+1}  
int main()   {cout <<S(1)}
```

- A. $main() \rightarrow S(1) \rightarrow S(0)$ B. $S(0) \rightarrow S(1) \rightarrow main()$
C. $main() \rightarrow S(0) \rightarrow S(1)$ D. $S(1) \rightarrow S(0) \rightarrow main()$

(14) 【2016】设有如下图所示的火车车轨, 入口到出口之间有 n 条轨道, 列车的行进方向均为从左至右, 列车可驶入任意一条轨道。现有编号为 1~9 的 9 列列车, 驶入的次序依次是 8、4、2、5、3、9、1、6、7。若期望驶出的次序依次为 1~9, 则 n 至少是 ()。



A. 2

B. 3

C. 4

D. 5

(15) 【2017】下列关于栈的叙述中,错误的是()。

*I.*采用非递归方式重写递归程序时必须使用栈

*II.*函数调用时,系统要用栈保存必要的信息

*III.*只要确定了入栈顺序,即可确定出栈次序

*IV.*栈是一种受限的线性表,允许在其两端进行操作

A. 仅 *I*

B. 仅 *I*、*II*、*III*

C. 仅 *I*、*III*、*IV*

D. 仅 *II*、*III*、*IV*

4.4.3 数组与矩阵压缩存储-习题精编(答案见原书页码 P106)

(1) 用足够容量的一维数组 B 对 $n \times n$ 阶对称矩阵 A 进行压缩存储, 若 B 中只存储对称矩阵 A 的下三角元素, 则 $A[i, j] (i < j, 0 \leq i, j \leq n-1)$ 存储在 B 中对应的元素为 ()。

- A. $B[j * n/2 + i]$ B. $B[i * (i+1)/2 + j]$ C. $B[j * (j+1)/2 + i]$ D. $B[i * n/2 + j]$

(2) 下三角矩阵 $A(n \times n)$ 按列优先顺序压缩在数组 $S[(n+1) * n/2]$ 中, 若非零元素 $a_{ij} (0 \leq i, j < n)$ 存放在 $S[k]$ 中, 则 i, j, k 的关系为 ()。

A. $k = i * n + j$

B. $k = (2n - j + 1) * j/2 + i - 1$

C. $k = (i + 1) * i/2 + j$

D. $k = (2n - j + 1) * j/2 + i - j$

(3) 若将 6×6 的上三角矩阵 A (下标从 1 起) 的上三角元素按行优先存储在一维数组 b 中, 且 $b[1] = A_{11}$, 那么 A_{35} 在 b 的下标是 ()。

A. 12

B. 13

C. 14

D. 15

(4) 稀疏矩阵的压缩存储缺点在于()。

- A. 无法得到矩阵的维数信息
- B. 无法根据行列号查找矩阵元素
- C. 无法随机存取
- D. 使矩阵的逻辑关系变得更复杂

(5) 若以行优先顺序存储三维数组 $A[80][20][40]$, 其中元素 $A[0][0][0]$ 所在地址为 0, 且每个元素占 4 个存储单元, 则 $A[20][10][30]$ 的地址为 ()。

A. 65720

B. 65724

C. 65716

D. 65728

4.4.4 数组与矩阵压缩存储-真题演练(答案见原书页码 P106)

(6) 【2016】有一个 100 阶的三对角矩阵 M , 其元素 $m_{i,j}$ ($1 \leq i \leq 100, 1 \leq j \leq 100$) 按行优先次序压缩存入下标从 0 开始的一维数组 N 中。元素 $m_{30,30}$ 在 N 中的下标是 ()。

A. 86

B. 87

C. 88

D. 89

(7) 【2017】适用于压缩存储稀疏矩阵的两种存储结构是()。

A. 三元组表和十字链表

B. 三元组表和邻接矩阵

C. 十字链表和二叉链表

D. 邻接矩阵和十字链表

(8) 【2018】设有一个 12×12 的对称矩阵 M , 将其上三角部分的元素 $m_{i,j} (1 \leq i \leq j \leq 12)$ 按行优先存入 C 语言的一维数组 N 中, 元素 $m_{6,6}$ 在 N 中的下标是 ()。

A. 50

B. 51

C. 55

D. 66

(9) 【2020】将一个 10×10 对称矩阵 M 的上三角部分的元素 $m_{i,j} (1 \leq i \leq j \leq 10)$ 按列优先存入 C 语言的一维数组 N 中, 元素 $m_{7,2}$ 在 N 中的下标是 ()。

A. 15

B. 16

C. 22

D. 23

(10) 【2021】已知二维数组 A 按行优先方法存储, 每个元素占用 1 个存储单元, 若元素 $A[0][0]$ 的存储地址为 100, $A[3][3]$ 的存储地址为 220, 则元素 $A[5][5]$ 的存储地址是()。

A. 295

B. 300

C. 301

D. 306

第 5 章树与二叉树

5.1.4 树的基本概念-习题精编(答案见原书页码 P114)

(1) 以下关于树的描述, 错误的是()。

- A. 非叶结点有可能没有孩子
- B. 在一棵树中加一条边会形成一个环
- C. 一棵树中并非每个结点都有双亲结点
- D. 一棵包含一个结点的树高度是 1

(2) 一棵具有 n 个结点, 度为 4 的树, 满足 ()。

- A. 树高至多是 $n - 3$
- B. 至少在某一层上正好有 4 个结点
- C. 第 i 层上至多有 4^i 个结点
- D. 树高至多是 $n - 2$

(3) 度为4、高度为 h 的树满足()。

- A. 至少有 $h+3$ 个结点
- B. 至多 4^h-1 个结点
- C. 第 i 层至多有 4^i 个结点
- D. 至少在某层上正好4个结点

(4) 三叉树中有 1、2、3 个子树的结点数为 x 、 y 、 z , 其叶结点数目是 ()。

A. $x + y + z$

B. $2y + 1$

C. $y + 2z + 1$

D. $2z + 1$

(5) 包含 n 个结点的树的所有结点的度数之和是_____。

(6) 度为 4 的树且结点数是 56, 其最小高度为 _____。

5.1.5 树的基本概念-真题演练(答案见原书页码 P114)

(9) 【2010】在一棵度为 4 的树中,若有 20 个度为 4 的结点, 10 个度为 3 的结点, 1 个度为 2 的结点, 10 个度为 1 的结点,则树 T 的叶结点个数是 ()。

A. 41

B. 82

C. 113

D. 122

5.2.5 二叉树基础-习题精编(答案见原书页码 P122)

(1) 若二叉树非空, 具有 n 个结点且深度也是 n 的二叉树有 () 种。

A. $2n$

B. 2^n

C. 2^{n-1}

D. $2^n - 1$

(2) 在一棵存储于数组的二叉树中, 根结点的索引号为 0, 则索引号为 n 的结点的孩子结点 (若存在) 的索引号是 ()。

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

(3) 下列说法中, 正确的是()。

- A. 在不考虑空树的情况下, 一颗二叉树就是一颗度为2的有序树
- B. 含有 n 个结点的二叉树的高度为 $\lfloor \log_2 n + 1 \rfloor$
- C. 完全二叉树中没有右孩子的结点不一定是叶结点, 没有左孩子的结点必定是叶结点。
- D. 完全二叉树中所有叶结点位于同一层

(4) 以下说法中, 正确的是()。

- A. 完全二叉树中, 一个叶结点的左侧叔结点有孩子结点, 则该左侧叔结点一定不是叶子结点
- B. 任何一棵非空二叉树内 $n_0 = n_2 - 1$ (n_0 为叶子节点个数, n_2 为度为 2 的节点个数)
- C. 除了完全二叉树外, 其它任何二叉树都不适合顺序存储结构。
- D. 结点按完全二叉树层序编号的二叉树中 (数组下标编号是从 0 开始的), 第 i 个结点的左孩子 (若存在) 的编号为 $2i$

(5) 设二叉树共 $2n$ 个结点, 且 $m < n$, 则其中的结点数量不可能存在的情况是 ()。

A. n 个度为 0

B. $2m$ 个度为 0

C. $2k$ 个度为 1, k 取合理值

D. $2m - 1$ 个度为 1

(6) 假设一棵二叉树 T 的结点个数为 31, 则下列说法中正确的是 ()。

I. T 的最小高度为 5, 最大高度为 31

II. T 中最少有 1 个叶子结点, 最多有 15 个叶子结点

III. 若 T 中分支结点的度均为 1, 则 T 的所有可能的形态共有 2^{31} 种

IV. 若 T 中分支结点的度均为 2, 则 T 必为完全二叉树, 可能为满二叉树

A. *I*

B. *I, II*

C. *I, II, IV*

D. *II, III, IV*

(7) 一棵有 $n(n \geq 1)$ 个结点的二叉树 T 采用三叉链表存储结点, 其中每个结点包含三个指针, 分别指向其左孩子、右孩子以及双亲结点 (若不存在则置为空), 则下列说法中正确的是 ()。

I. T 中空指针的数量为 $n + 2$

II. 所有度为 2 的结点均被三个指针指向 (只考虑指向孩子结点和指向双亲结点的指针)

III. 每个叶子结点均被一个指针所指向 (只考虑指向孩子结点的指针和指向双亲结点的指针)

A. I

B. I, II

C. I, III

D. II, III

(8) 深度为 h 的完全二叉树最少有 () 个结点。

A. 2^{h+1}

B. 2^h

C. 2^{h-1}

D. $2^h - 1$

(9) 二叉树的根结点在第 1 层,总共有 125 个结点,那么在理想情况下,该二叉树的第 7 层的结点总数最多是()。

A. 62

B. 64

C. 63

D. 不存在第七层

(10) 拥有 56 个结点的三叉树可能存在的最小高度是_____.

(11) 一颗二叉树共有 4 个结点, 那么它的可能有 _____ 种形态。

(12) 有 1001 个结点的完全二叉树的叶子结点有 _____ 个。

(13) 假设高度为 h 的二叉树上只有度为 0 和度为 2 的结点, 则这棵树上的结点数至少为 ()。

A. $2h$

B. $2h - 1$

C. $2h + 1$

D. $h + 1$

(14) 三叉树中,有度为3的结点2个,度为2的结点2个,度为1的结点4个,则度为0的结点有 _____ 个。

(15) 在一个用数组表示的完全二叉树中,根结点下标为1,那么下标为17和19结点的最近公共祖先的下标是()。

A. 1

B. 2

C. 4

D. 8

(16) 若二叉树的总结点数为 98,其中仅有一个孩子的结点数为 48,则该树的叶结点数是()。

A. 25

B. 50

C. 不确定

D. 这样的树不存在

(17) 一棵完全二叉树共 626 个结点,则叶子结点的数目为()。

A. 311

B. 312

C. 313

D. 314

(18) 已知二叉树有 50 个叶结点, 则该二叉树的总结点数至少为 _____。

(19) 在某棵树中, 结点 M 和 N 是结点 P 的第 i 和 $i+1$ 个孩子, 则在这棵树的二叉树表示中, 结点 M 与 N 的关系是()。

- A. M 、 N 具有同一双亲
- B. M 是 N 的左孩子
- C. N 是 M 的左孩子
- D. N 是 M 的右孩子

(20) 具有 $2n$ 个结点的完全二叉树, 含有 _____ 个度为 1 的结点, _____ 个度为 2 的结点。

(21) n 个结点的 k 叉树 ($k \geq 2$) 的 k 叉链表中有 _____ 个空指针。

(22) 一个具有 2049 个结点的二叉树的高 h 为()。

A. 12

B. 11

C. 12 至 2049 之间

D. 11 至 2048 之间

(23) 已知一个二叉树中叶子数为 50, 仅有一个孩子的结点数为 30, 则总结点数为()。

A. 81

B. 129

C. 130

D. 无法确定

(24) 第7层有10个叶结点的完全二叉树不可能有()个结点。

A. 73

B. 234

C. 235

D. 236

5.2.6 二叉树基础-真题演练(答案见原书页码 P122)

(27) 【2009】已知一棵完全二叉树的第6层(设根为第1层)有8个叶结点,则该完全二叉树的结点个数最多是()。

A. 39

B. 52

C. 111 三

D. 119

(28) 【2011】若一棵完全二叉树有 768 个结点,则该二叉树中叶结点的个数是 ()。

A. 257

B. 258

C. 384

D. 385

(29) 【2011】 已知一棵有 2011 个结点的树,其叶结点个数为 116,该树对应的二叉树中无右孩子的结点个数是 ()。

A. 115

B. 116

C. 1895

D. 1896

(30) 【2018】设一棵非空完全二叉树 T 的所有叶结点均位于同一层, 且每个非叶结点都有 2 个子结点。若 T 有 k 个叶结点, 则 T 的结点总数是 ()。

A. $2k - 1$

B. $2k$

C. k^2

D. $2^k - 1$

(31) 【2020】对任意高度为5且有10个结点的二叉树,若采用顺序存储结构保存,每个结点占1个存储单元(仅存放结点的数据信息),则存放该二叉树需存储单元数量至少是()。

A. 31

B. 16

C. 15

D. 10

5.3.5 二叉树的遍历-习题精编(答案见原书页码 P140)

(1) 若二叉树有两个结点 p 和 q , 对该树进行中序遍历时, p 在 q 的前面, 则 ()。

- A. p 是 q 的祖先 B. q 是 p 的祖先 C. p 在 q 的左边 D. q 在 p 的左边

(2) 若二叉树所含的结点个数大于 1, 且每个结点都仅有一个孩子, 则()是不可能发生的。

- A. 中序与先序相同 B. 层序和先序相同 C. 中序与后序相同 D. 层序和后序相同

(3) 已知 $|1|2|3|4|5|6|7|8|9|10|$ 是一棵完全二叉树的顺序存储结构, 问该二叉树的后序遍历结果是 ()。

A. 1,2,3,4,5,6,7,8,9,10

B. 1,2,4,8,9,5,10,3,6,7

C. 8,4,9,10,5,2,6,3,7,1

D. 8,9,4,10,5,2,6,7,3,1

(4) 对二叉树 T 中的某个结点 x , 它在先序遍历, 中序遍历, 后序遍历中的序号分别是 $pre(x), in(x), post(x)$, a 和 b 是 T 中的任意两个结点, 下列选项一定错误的是 ()。

- A. a 是 b 后代且 $pre(a) < pre(b)$
- B. a 是 b 的祖先且 $post(a) > post(b)$
- C. a 是 b 后代且 $in(a) < in(b)$
- D. a 在 b 的左边且 $in(a) < in(b)$

(5) 二叉树中 n 是 m 的祖先, 在 () 中, n 在 m 前面是不可能的。

A. 先序序列

B. 中序序列

C. 后序序列

D. 层序序列

(6) 若一个二叉树的先序序列是 $ABCDE$, 树深度不超过 3, 那么其可能的形态有 () 种。

A. 4

B. 6

C. 7

D. 8

(7) 一棵二叉树中若结点 A 有左孩子 B 、右孩子 C , 则在该二叉树的先序、中序、后序序列中, 下列说法正确的是()。

- A. 结点 A 一定排在结点 B 的前面
- B. 结点 A 一定排在结点 C 的前面
- C. 结点 B 一定排在结点 C 的前面
- D. 结点 C 可能排在结点 B 的前面

(8) 设 n, m 为一棵二叉树上的两个结点, 在中序遍历时, n 在 m 前的条件是 ()。

- A. n 在 m 右方 B. n 是 m 祖先 C. n 在 m 左方 D. n 是 m 子孙

(9) 某二叉树先序遍历序列: $ABCDEFG$, 后序遍历序列: $EDCGFBA$, 则 ()。

- A. 结点 C 可能是结点 A 的孩子
- B. 结点 E 可能是内部结点
- C. 结点 F 可能是结点 G 的兄弟
- D. 结点 A 必定仅有一个孩子结点

(10) 通过 () 可找到一个结点到其祖先的路径。

A. 先序遍历

B. 中序遍历

C. 后序遍历

D. 层次遍历

(11) 在二叉树的先序序列、中序序列和后序序列中,所有叶子结点的先后顺序()。

- A. 都不相同
- B. 完全相同
- C. 先序和中序相同, 而与后序不同
- D. 中序和后序相同, 而与先序不同

(12) 先序为 ABC , 后序为 BCA 的二叉树共有 ()。

A. 1 棵

B. 2 棵

C. 3 棵

D. 4 棵

(13) 一棵非空的二叉树的先序遍历序列与后序遍历序列正好相反, 则该二叉树一定满足()。

- A. 所有的结点均无左孩子
- B. 所有的结点均无右孩子
- C. 只有一个叶结点
- D. 是任意一棵二叉树

(14) 设结点 X 和 Y 是二叉树中任意的两个结点。在该二叉树的先序遍历序列中 X 在 Y 之前, 而在其后序遍历序列中 X 在 Y 之后, 则 X 和 Y 的关系是 ()。

- A. X 是 Y 的左兄弟 B. X 是 Y 的右兄弟 C. X 是 Y 的祖先 D. X 是 Y 的后代

(15) 若二叉树的先序序列中 a 结点在 b 前, 中序序列中 b 结点在 a 结点之前, 则 ()。

- A. 结点 a 和 b 分别在某结点的右子树和左子树中
- B. 结点 b 在结点 a 的右子树中
- C. 结点 b 在结点 a 的左子树中
- D. 结点 a 和 b 分别在某结点的两棵不同的子树中

(16) (多选) 下列的中序遍历序列能和先序遍历 1234567 构成一颗二叉树的是()。

A. 3214567

B. 1234567

C. 4136527

D. 1463527

(17) 通过()的序列组合, 不能还原出唯一的二叉树。

- A. 层次序列和中序序列
- B. 先序序列和中序序列
- C. 后序序列和中序序列
- D. 先序序列和后序序列

(18) 由后序序列: $DABEC$ 和中序序列: $DEBAC$, 可以唯一确定一棵二叉树 T 。则 T 的先序序列为 ____
____。

(19) 由先序序列: $ABCDEF$ 和中序序列: $CBAEDF$, 可以唯一确定一棵二叉树 T 。则 T 的后序序列为 _____。

(20) 由层次序列: $ABCDEF$ 和中序序列: $BADCFE$, 可以唯一确定一棵二叉树 T 。则 T 的先序序列为 _____。

(21) 一棵二叉树的先序遍历序列和其后序遍历序列正好相反, 则该二叉树一定是()。

- A. 空树或者只有一个结点
- B. 完全二叉树
- C. 二叉排序树
- D. 高度等于其结点数

(22) 若二叉树的中序序列是 $HFIEJKG$, 先序序列是 $EFHIGJK$, 那么按 NRL 次序遍历该二叉树, 得到的序列的第 2 个结点是_____。

(23) 中缀表达式: $A + B * C - D / E$ 对应的前缀表达式是_____。

(24) 在一棵非空二叉树的中序遍历序列中,根结点的左边()。

- A. 有右子树上的所有结点
- B. 只有右子树上的部分结点
- C. 只有左子树上的部分结点
- D. 有左子树上的所有结点

(25) 某二叉树的先序遍历序列和中序遍历序列正好相反, 则该二叉树一定是()。

- A. 任一结点无左孩子
- B. 任一结点无右孩子
- C. 只有一个根结点
- D. 不存在

(26) 已知一二叉树的先序遍历的结果是 ABC . 则以下哪个序列是不可能的中序遍历结果 ()。

A. ABC

B. BAC

C. CAB

D. 以上都有可能

(27) 对二叉树的结点从1开始连续编号,要求每个结点的编号大于其左、右孩子的编号,同一结点的左右孩子中,其左孩子的编号小于其右孩子的编号,可采用()实现编号。

- A. 先序遍历 B. 中序遍历 C. 后序遍历 D. 从根开始层序遍历

(28) 一棵二叉树的先序遍历结果为 $abcdef$, 中序遍历结果为 $cbaedf$, 则后序遍历结果为 ()。

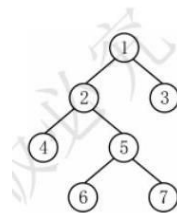
- A. $cbefda$ B. $fedcba$ C. $cbedfa$ D. 不定

(29) (多选) 一棵二叉树采用二叉链表表示, 若要采用递归的方法将其所有结点的左右子树交换位置, 则采用下列 () 遍历的方法比较合适。

- A. 先序 B. 中序 C. 后序 D. 层序

5.3.6 二叉树的遍历-真题演练(答案见原书页码 P140)

(45) 【2009】 给定二叉树如右图所示。设 N 表二叉树的根, L 表根结点的左子树, R 表根结点的右子树。若遍历后的结点序列是 3,1,7,5,6,2,4,则其遍历方式是 ()。



A. LRN

B. NRL

C. RLN

D. RNL

(46) 【2011】若一棵二叉树的先序遍历序列和后序遍历序列分别为 1,2,3,4 和 4,3,2,1, 则该二叉树的中序遍历序列不会是()。

A. 1,2,3,4

B. 2,3,4,1

C. 3,2,4,1

D. 4,3,2,1

(47) 【2012】若一棵二叉树的先序遍历序列为 $aebdc$, 后序遍历序列为 $bcdea$, 则根结点的孩子结点 ()。

- A. 只有 e B. 有 e 、 b C. 有 e 、 c D. 无法确定

(48) 【2015】先序序列为 a, b, c, d 的不同二叉树的个数是 ()。

A. 13

B. 14

C. 15

D. 16

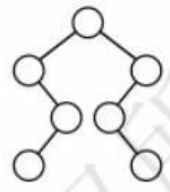
(49) 【2017】 已知一棵二叉树的树形如右图所示, 其后序序列为 *eachdbgf* 树中与结点 *a* 同层的结点是 ()。

A. *c*

B. *d*

C. *f*

D. *g*



(50) 【2017】要使一棵非空二叉树的先序序列与中序序列相同, 其所有非叶结点须满足()。

- A. 只有左子树
- B. 只有右子树
- C. 结点的度均为 1
- D. 结点的度均为 2

5.4.4 线索二叉树-习题精编(答案见原书页码 P160)

(1)(多选) 下列关于先序线索树中查找结点的先序后继的说法中, 错误的是()。

- A. 当指定结点不是叶结点时,若指定结点有左孩子,则左孩子就是它的先序后继,若指定结点没有左孩子, 则右孩子是它的先序后继
- B. 当指定结点是叶结点时, 若指定结点是某结点 X 的左子树中先序遍历序列的最后一个结点,且结点 X 有右孩子, 则该指定结点的先序后继就是结点 X 的右孩子
- C. 当指定结点是叶结点时, 若指定结点虽然是某结点 X 的左子树中先序遍历序列的最后一个结点, 但是该结点 X 没有右孩子, 则指定结点没有先序后继
- D. 当指定结点是叶结点时,若指定结点不是任何结点的左子树中先序遍历序列的最后一个结点,则指定结点先序后继是根结点

(2) n 个结点的线索二叉树上含有的线索数为()。

A. $2n$

B. $n - 1$

C. $n + 1$

D. n

(3) 左子树为空的二叉树在先序线索化后, 其空的链域的个数是()。

A. 不确定

B. 0 个

C. 1 个

D. 2 个

(4) 在线索二叉树中, 下列说法不正确的是()。

- A. 在中序线索树中,若某结点有右孩子,则其后继结点是它的右子树的最左下结点
- B. 在中序线索树中,若某结点有左孩子,则其前驱结点是它的左子树的最右下结点
- C. 线索二叉树是利用二叉树的 $n + 1$ 个空指针来存放结点的前驱和后继信息的
- D. 每个结点通过线索都可以直接找到它的前驱和后继

(5) 二叉树在线索化后, 仍不能有效求解的问题是()。

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

(6) 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 的前驱为 ()。

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

(7) 下列遍历方式中, 仍需要栈的支持的是()。

- A. 先序线索树 B. 中序线索树 C. 后序线索树 D. 所有线索树

(8) 二叉树的先序序列和后序序列正好相反, 则该二叉树一定是()。

- A. 空或只有一个结点
- B. 高度等于其结点数
- C. 任一结点无左孩子
- D. 任一结点无右孩子

(9) 设一个二叉树的先序和中序遍历结果分别是 $ABCDFE$ 和 $CBAFDE$, 那么在该树对应的二叉中序线索树中, 下列哪些结点的右链都是线索 ()。

- A. C 和 B B. B 和 A C. C 和 D D. D 和 F

(10) 高度为 H 的后序线索二叉树中, p 是其中的一个结点, q 是 p 的左孩子, p 的右子树高度为 H_1 ,请问 q 到 q 的后继结点的连线路径上最多经过_____个结点(不含路径的两端点)。

(11) 对于一个线索化的二叉树, 其中 p 所指结点无左子树的充要条件为()。

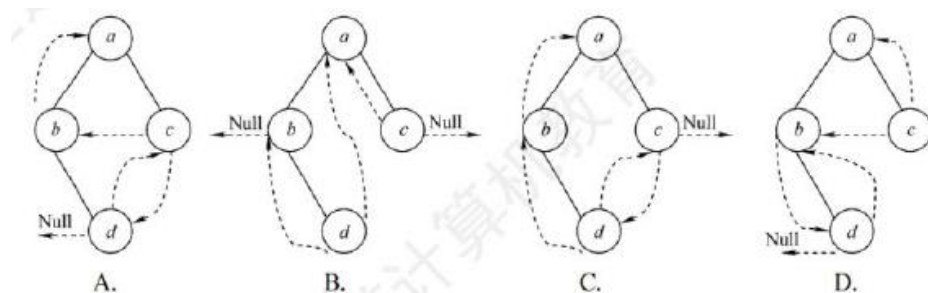
- A. $p \rightarrow lChild == NULL$
- B. $p \rightarrow ltag == 1$
- C. $p \rightarrow ltag == 1$ 且 $p \rightarrow lChild == NULL$
- D. 以上皆错

(12) 一棵具有 n 个非叶子结点完全二叉树的线索树, 含有 () 条线索。

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

5.4.5 线索二叉树-真题演练(答案见原书页码 P160)

(13) 【2010】下列线索二叉树中(虚线表示线索), 符合后序线索树定义的是 ()。



(14) 【2013】 X 是后序线索二叉树的叶结点, 且 X 存在左兄弟结点 Y , 则 X 右线索指向()。

- A. X 的父结点
- B. Y 为根的子树的最左下结点
- C. X 的左兄弟结点 Y
- D. Y 为根的子树的最右下结点

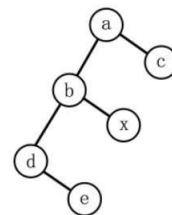
(15) 【2014】若对如右图的二叉树进行中序线索化, 则结点 x 的左、右线索指向的结点分别是 ()。

A. e 、 c

B. e 、 a

C. d 、 c

D. b 、 a



5.5.4 树与森林-习题精编(答案见原书页码 P169)

(1) n 个结点的 m 叉树转化成对应的二叉树, 所需的存储资源比未转化前用定长结点存储节省了 () 个指针。

A. $(m-2)n$

B. mn

C. $(m-1)n$

D. m

(2) 设一个森林 G 有 5 个结点, 3 棵树. 则其边的条数是 ()。

A. 2

B. 3

C. 4

D. 5

(3) 下列说法中, 正确的是()。

I. 对于有 n 个结点的任意形态的二叉树的高度一定是 $\log_2 n$

II. 完全二叉树中, 若某个结点没有左孩子, 则它一定是叶结点

III. 高度为 h 的完全二叉树转换成对应的森林后, 其所含的树的个数一定是 h

IV. 一棵树或森林中的叶子数一定等于与其转化后对应的二叉树的叶子数

A. *I* 和 *III*

B. *IV*

C. *I* 和 *II*

D. *II*

(4) 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

(5) 设森林 F 中有 3 棵树, 第一、第二、第三棵树的结点个数分别为 M_1, M_2 和 M_3 。与森林 F 对应的二叉树根结点的右子树上的结点个数是 ()。

- A. M_1 B. $M_2 + M_2$ C. M_3 D. $M_2 + M_3$

(6) 设森林 F 对应的二叉树为 B , 它有 m 个结点, B 的根为 p , p 的右子树结点个数为 n , 森林 F 中第一棵树的结点个数是 ()。

A. $m - n$

B. $m - n - 1$

C. $n + 1$

D. 条件不足, 无法判断

(7) 将森林 $T = (T_1, T_2, \dots, T_m)$ 转化为二叉树 BT 的过程如下: 若 m 为 0, 则 BT 为空, 若 m 不为 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 的子树森林转换成 BT 的左子树; 将 (T_2, T_2, \dots, T_m) 转换成 BT 的右子树
- C. 将子树 T_1 的根作为 BT 的根; 将 T_1 的左子树森林转换成 BT 的左子树; 将 T_1 的右子树森林转换为 BT 的右子树; 其他以此类推
- D. 将森林 T 的根作为 BT 的根; 将 (T_1, T_2, \dots, T_m) 转化为该根下的结点, 得到一棵树, 然后将这棵树再转化为二叉树 BT

(8) 设 F 是一个森林, B 是由 F 变换来的二叉树。若 F 中有 n 个非终端结点, 则 B 中右指针域为空的结点有 () 个。

A. $n - 1$

B. n

C. $n + 1$

D. $n + 2$

(9) 中序序列为 $BDAECF$ 、后序序列为 $DBEFCA$ 的二叉树对应的森林包含 () 棵树。

A. 1

B. 2

C. 3

D. 4

(10) 将树 T 转换成对应的二叉树 B 后, 树 T 中的一个结点 X , 在二叉树 B 中变成了其父结点的右孩子。则下列结论中正确的是 ()。

- A. T 中 X 是其双亲结点的第一或第二个孩子
- B. T 中, X 必定有右兄弟
- C. T 中, X 必定是一个非叶子结点
- D. T 中, X 一定有左兄弟

(11) 在森林的二叉树表示中, 结点 M 和结点 N 是同一父结点的左孩子和右孩子, 则 ()。

- A. M 和 N 有同一双亲
- B. M 和 N 可能无公共祖先
- C. M 是 N 的儿子
- D. M 是 N 的左兄弟

(12) 如果森林 F 采用“孩子—兄弟”表示法对应的二叉树是 16 个结点的完全二叉树, 问: 森林 F 中树的数目和最大树的结点个数分别为()。

A. 2 和 8

B. 2 和 9

C. 4 和 8

D. 4 和 9

(13) 对于一个共有 n 个结点、 K 条边的森林, 共有 () 棵树。

A. $n - K$

B. $n - K + 1$

C. $n - K - 1$

D. 不确定

(14) 一个森林有三棵树, 分别有 M_1, M_2, M_3 个结点, 则该森林对应的二叉树的根结点的右子树有 ____
____ 个结点。

(15) 将森林 F 转换为对应的二叉树 T , F 中任何一个没有右兄弟的结点, 在 T 中 ()。

- A. 没有左子树 B. 没有右子树 C. 没有左和右子树 D. 以上都不对

(16) 在下列选项中, 不能作为树的存储形式是()。

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

5.5.5 树与森林-真题演练(答案见原书页码 P169)

(17) 【2009】将森林转换为对应的二叉树,若在二叉树中,结点 u 是结点 v 的父结点的父结点,则在原来的森林中, u 和 v 可能具有的关系是()。

I.父子关系 *II*.兄弟关系 *III*. u 的父结点与 v 的父结点是兄弟关系

- A. 只有 *II* B. *I* 和 *II* C. *I* 和 *III* D. *I*、*II* 和 *III*

(18) 【2014】将森林 F 转换为对应的二叉树 T , F 中叶结点的个数等于 ()。

- A. T 中叶结点的个数
- B. T 中度为 1 的结点个数
- C. T 中左孩子指针为空的结点个数
- D. T 中右孩子指针为空的结点个数

(19) 【2019】若将一棵树 T 转化为对应的二叉树 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. 按层遍历

(20) 【2020】已知森林 F 及与之对应的二叉树 T , 若 F 的先根遍历序列是 $abcdef$, 后根遍历序列是 $badfec$, 则 T 的后序遍历序列是 ()。

A. $badfec$

B. $bdfeac$

C. $bfedca$

D. $fedcba$

(21) 【2021】某森林 F 对应的二叉树为 T , 若 T 的先序遍历序列是 $abdcegf$, 中序遍历序列是 $bdaegcf$, 则 F 中树的棵数是 ()。

A. 1

B. 2

C. 3

D. 4

5.6.3 树与二叉树的应用-习题精编(答案见原书页码 P182)

(1) 下列关于并查集的说法正确的是 ()。

- A. 并查集可用于实现 *Prim* 算法
- B. 通过路径优化后的并查集在最坏情况下的高度仍是 $O(n)$
- C. *Find* 操作返回集合元素个数的相反数,它用来当作代表某集合的标志
- D. 并查集基于双亲表示法

(2) 下列关于并查集的说法正确的是 ()。

- A. 如果同时使用路径压缩算法和 *Union* 优化算法的话, 对有 n 个元素的并查集上进行 m 次查询和合并操作, 其时间复杂度是 $O(m \log_2 n)$
- B. 并查集不能处理是否存在环路的问题
- C. 并查集可以用来验证树的边数 $+1 =$ 结点数的规律
- D. 并查集可以用来计算两个结点间的路径长度

(3) 下列不可能是某哈夫曼树的结点个数的是()。

A. 803

B. 308

C. 83

D. 以上皆错

(4) 在共有 n 个叶子结点哈夫曼树内, 非叶结点的总数是 ()。

A. $n - 1$

B. n

C. $n + 1$

D. $2n - 1$

(5) 设某哈夫曼树内有 199 个结点, 则该哈夫曼树中有 () 个叶子结点。

A. 99

B. 100

C. 101

D. 102

(6) 若用 4、5、6、7、8 作叶结点权值构建哈夫曼树, 其 WPL 是_____。

(7) 下列编码中,()不是前缀码。

A. 00,01,10,11

B. 0,1,00,11

C. 0,10,110,111

D. 10,110,1110,1111

(8) 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, 则还最多可对 () 个字符进行编码。

A. 2

B. 3

C. 4

D. 5

(9) 哈夫曼树总共有 107 个内部结点, 那么其能表示() 个不同的码字。

A. 215

B. 108

C. 109

D. 107

(10) 以下对于哈夫曼树的说法中,错误的是()。

- A. 同一组权值相应的构造出来的哈夫曼树可能不是唯一的
- B. 权值越小的结点离根结点越远
- C. WPL 最小的树不一定就是按哈夫曼树构造方法构造出来的
- D. 度为 m 的哈夫曼树中除度为 0 和 m 的结点外, 还可以有度为 1 到 $m-1$ 的结点

(11) 如果度为 m 的哈夫曼树内, 其叶子结点个数是 n , 那么其非叶子结点的个数是 ()。

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

(12) N 个不同权值的关键字对应的哈夫曼树的结点总数是 ()。

A. $N - 1$

B. $2N - 1$

C. $2N$

D. $2N + 1$

(13) 下列编码不是哈夫曼编码的是 ()。

A. 111, 110, 10, 01, 00

B. 000, 001, 010, 011, 1

C. 110, 11, 10, 1, 0

D. 000, 001, 01, 10, 11

(14) 假设哈夫曼二叉树中只有度为 0 或 2 的结点,根据频率为 5 个字符设计的哈夫曼编码,不可能的是 ()。

A. 000,001,010,011,1

B. 0000,0001,001,01,1

C. 000,001,01,10,11

D. 00,100,101,110,111

(15) 含有 10 个元素的并查集, 按 $1-2$ 、 $3-4$ 、 $5-6$ 、 $7-8$ 、 $0-9$ 、 $1-8$ 、 $9-5$ 、 $1-9$ 的顺序进行查找合并操作, 最终得到 () 个不同的集合。

A. 1

B. 2

C. 3

D. 4

(16) 设一段文本中包含对象 a, b, c, d, e , 其出现次数相应为 3, 2, 5, 1, 1, 则经过哈夫曼编码后, 该文本所占总位数为()。

A. 40

B. 36

C. 25

D. 12

(17) 在为 n 个对象设计最优编码时, 如果相应的哈夫曼树有 67 个结点, 则可知 n 的值为()。

A. 22

B. 33

C. 34

D. 不确定

(18) 从权值分别为 9, 2, 5, 7 的四个叶子结点构造一棵哈夫曼 (*Huffman*) 树, 则该树的带权路径长度 WPL 为 ()。

A. 23

B. 37

C. 44

D. 46

(19) 有 6 个字符出现的频率分别是 2、1、4、5、7、3, 如果采用哈夫曼方法进行最优编码, 那么字符编码长度最长的位数是 ()。

A. 3

B. 4

C. 5

D. 6

(20) 下面关于 *Huffman* 树的说法中错误的是 ()。

- A. 对应一组权重构造出来的 *Huffman* 树一般不是唯一的
- B. *Huffman* 树具有最小的带权路径长度
- C. *Huffman* 树中没有度为 1 的结点
- D. *Huffman* 树中除度为 1 的结点外,还有度为 2 的结点和叶结点

(21) 有 4 叉哈夫曼树, 结点与其四个孩子结点的路径分别编码为 00、01、10、11。现对关键字序列 1、1、2、3、5、8、13、21 构造 4 叉哈夫曼树并进行编码, 下列说法中正确的是 ()。

- A. 最小带权路径长度为 108
- B. 关键字 2 对应的编码长度为 6
- C. 编码长度为 6 的关键字有 4 个
- D. 010000, 010001, 0101, 0110, 0111, 00, 10, 11 是一个合法的编码序列

5.6.4 树与二叉树的应用-真题演练(答案见原书页码 P182)

(25) 【2010】对 $n(n \geq 2)$ 个权值均不相同的字符构造哈夫曼树。下列关于该哈夫曼树的叙述中，错误的是 ()。

- A. 该树一定是一棵完全二叉树
- B. 树中一定没有度为 1 的结点
- C. 树中两个权值最小的结点一定是兄弟结点
- D. 树中任一非叶结点的权值一定不小于下一层任一结点的权值

(26) 【2013】 已知三叉树 T 中 6 个叶结点的权分别是 2、3、4、5、6、7, T 的带权 (外部) 路径长度最小是 ()。

A. 27

B. 46

C. 54

D. 56

(27) 【2014】 5 个字符有如下 4 种编码方案, 不是前缀编码的是 ()。

A. 01, 0000, 0001, 001, 1

B. 011, 000, 001, 010, 1

C. 000, 001, 010, 011, 100

D. 0, 100, 110, 1110, 1100

(28) 【2015】下列选项给出的是从根分别到达两个叶结点路径上的权值序列,能属于同一棵哈夫曼树的是()。

- A. 24,10,5 和 24,10,7
- B. 24,10,5 和 24,12,7
- C. 24,10,10 和 24,14,11
- D. 24,10,5 和 24,14,6

(29) 【2017】 已知字符集 $\{a, b, c, d, e, f, g, h\}$, 若各字符的哈夫曼编码依次是 0100、10、0000、0101、001、011、11、0001, 则编码序列 0100011001001011110101 的译码结果是 ()

A. *acgabfh*

B. *adbagbb*

C. *afbeagd*

D. *afeefgd*

(30) 【2018】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

(31) 【2019】对 n 个互不相同的符号进行哈夫曼编码。若生成的哈夫曼树共有 115 个结点, 则 n 的值是 ()。

A. 56

B. 57

C. 58

D. 60

(32) 【2021】若某二叉树有 5 个叶子结点, 其权值分别为 10、12、16、21、30, 则其最小的带权路径长度 (WPL) 是 ()。

A. 89

B. 200

C. 208

D. 289

第 6 章图

6.1.3 图的基本概念-习题精编(答案见原书页码 P194)

(1) 无向图 $G = (V, E)$, 若图 G 的顶点数和边数相等, 则图 G 一定是 ()。

- A. 连通图 B. 非连通图 C. 无环的图 D. 有环的图

(2) 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中的极大连通子图
- D. 图的遍历就是从图中某一顶点出发并访问图中其余顶点

(3) 以下关于图的叙述中, 正确的是()。

- A. 强连通有向图的任何顶点到其他所有顶点都有弧
- B. 图的任意顶点的入度等于出度
- C. 有向完全图一定是强连通有向图
- D. 有向图的边集的任意子集和顶点集的任意子集均可构成原有向图的子图

(4) 若图 G 为非连通无向图, 含有 15 条边, 则图 G 至少有 () 个顶点。

A. 6

B. 7

C. 8

D. 9

(5) 强连通有向图 G_1 和连通无向图 G_2 都含有 n 个顶点, 则 G_1 和 G_2 最少边条数分别为()。

A. $n, n-1$

B. $n(n-1), n-1$

C. n, n

D. $n(n-1), n$

(6) 无向图 G 有 100 条边, 度为 9 和度为 11 的顶点均各有 10 个, 其余都是度为 3 的顶点, 则图 G 共有 () 个顶点。

A. 20

B. 23

C. 26

D. 29

(7) 若有向图 $G = (V, E)$ 的顶点数为 n 。则 G 每个顶点的入度和出度之和最大值可是 ()。

A. n

B. $n - 1$

C. $2n$

D. $2n - 2$

(8) 具有 6 个顶点的无向图, 当有 () 条边时能确保是一个连通图。

A. 8

B. 9

C. 10

D. 11

(9) 设有无向图 $G = (V, E)$ 和 $G' = (V', E')$, 若 G' 是 G 的生成树, 则下列不正确的是 ()。

I. G' 一定是 G 的子图

II. G' 为 G 的连通分量

III. G' 为 G 的无环子图

IV. G' 为 G 的极小连通子图且 $V' = V$

A. 只有 *IV*

B. *II*、*III*

C. *III*、*IV*

D. 只有 *II*

(10) 如果一个环具有 n 个顶点, 则该环含有 () 棵生成树。

A. n^2

B. n

C. $n - 1$

D. 1

(11) 无向图 $G = (V, E)$, $|V| = n$, $|E| = e$, 且图 G 是森林。那么该图 G 中必有 () 棵树。

A. n

B. e

C. $n - e$

D. 1

(12) 设无向图的顶点个数为 n , 则该图最多有 () 条边。

A. $n - 1$

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

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

D. 0

(13) 含有 n 个顶点的完全有向图的边数为 ()。

A. n^2

B. $n(n+1)$

C. $n/2$

D. $n(n-1)$

(14) 一个有 n 个结点的无向图, 最少有 () 个连通分量; 最多有 ()。

A. 0

B. 1

C. $n - 1$

D. n

(15) 在一个无向图中,所有顶点的度数之和等于所有边数的()倍;在一个有向图中,所有顶点的入度之和等于所有顶点出度之和的()倍。

A. $1/2$

B. 2

C. 1

D. 4

(16) 若一有向图有 6 个顶点, 11 条边, 问该图是稠密图还是稀疏图?()

- A. 稠密图 B. 稀疏图 C. 都是 D. 都不是

(17) 下列有关图的定义, 错误的是()。

- A. 无向图的所有顶点数之和可能不是边数的两倍
- B. 若存在一个含有 n 个顶点, 且边数最小值大于 $n-2$ 的图, 那么该图一定含有环
- C. 有向图所有顶点的入度与出度之和相等并且等于边数
- D. 如果从 u 到 v 根本不存在路径, 则可记该距离为无穷

6.1.4 图的基本概念-真题演练(答案见原书页码 P194)

(20) 【2009】下列关于无向连通图特性的叙述中, 正确的是 ()。

*I.*所有顶点的度之和为偶数 *II.*边数大于顶点个数减 1 *III.*至少有一个顶点的度为 1

- A. 只有 *I* B. 只有 *II* C. *I* 和 *II* D. *I* 和 *III*

(21) 【2010】若无向图 $G=(V,E)$ 中含有 7 个顶点, 要保证图 G 在任何情况下都是连通的, 则需要的边数最少是()。

A. 6

B. 15

C. 16

D. 21

(22) 【2017】已知无向图 G 含有 16 条边, 其中度为 4 的顶点个数为 3, 度为 3 的顶点个数为 4, 其他顶点的度均小于 3。图 G 所含的顶点个数至少是()。

A. 10

B. 11

C. 13

D. 15

6.2.6 图的存储-习题精编(答案见原书页码 P207)

(1) 以下有关图存储结构的描述中,不正确的是()。

- A. 如果不使用压缩存储的技术, 则某图采用邻接矩阵来进行存储, 该邻接矩阵所占用的存储空间大小与图的边数无关,只取决于图的顶点数大小
- B. 邻接矩阵既适用于无向图又适用于有向图的存储, 而邻接表一般只能用于有向图的存储
- C. 如果某有向图使用邻接矩阵存储, 且该邻接矩阵对角线以下的元素均为零元素, 那么该有向图的拓扑序列一定存在
- D. 因为用来表示无向图的邻接矩阵一定对称,所以只需要运用矩阵压缩存储技术对该邻接矩阵的上/下三角部分的矩阵元素进行存储

(2) 无向图 $G = (V, E)$ 用邻接矩阵表示, 且 $|V| = n, |E| = e$, 则在该邻接矩阵中取值为 0 的元素应该一共有 () 个。

A. e

B. $2e$

C. $n^2 - e$

D. $n^2 - 2e$

(3) 用邻接矩阵表示一张带权有向图 $G = (V, E)$, 顶点 $v_i \in V$, 则 v_i 的出度等于邻接矩阵中 (); v_i 的入度等于邻接矩阵中 ()。

- | | |
|-------------------------|-------------------------|
| A. 第 i 行非无穷的元素个数 | B. 第 i 列非无穷的元素个数 |
| C. 第 i 行非无穷且非 0 的元素个数 | D. 第 i 列非无穷且非 0 的元素个数 |

(4) 从邻接矩阵 $A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ 可以看出, 该图共有 (1) 个顶点; 若该图是有向图, 则该图共有 (2) 条弧; 若是无向图, 则共有 (3) 条边。

- | | | | | |
|----------|------|------|------|-----------|
| (1) A. 9 | B. 3 | C. 6 | D. 1 | E. 答案均不正确 |
| (2) A. 5 | B. 4 | C. 3 | D. 2 | E. 答案均不正确 |
| (3) A. 5 | B. 4 | C. 3 | D. 2 | E. 答案均不正确 |

(5) 有向图 $G = (V, E)$ 用邻接表表示, 且 $|V| = n, |E| = e$ 。顶点 $v \in V$, 那么移除图 G 中与 v 有关的所有边的时间复杂度应该为 ()。

- A. $O(n)$ B. $O(e)$ C. $O(n+e)$ D. $O(ne)$

(6) 对邻接表的叙述中,下面哪个说法是正确的()。

- A. 无向图的邻接表中, 第 i 个顶点的度为第 i 个链表中结点数的两倍
- B. 邻接表比邻接矩阵的操作更方便
- C. 邻接矩阵比邻接表的操作更方便
- D. 求有向图结点的度,必须遍历整个邻接表

(7) 如果邻接表中有奇数个边表结点, 则下列说法中正确的是()。

- A. 图中有奇数个结点
- B. 图中有偶数个结点
- C. 图为无向图
- D. 图为有向图

(8) 若有向图 $G = (V, E), |V| = n$, 用邻接矩阵 A 表示图 G 时, 顶点 V_i 的度是 ()。

A. $\sum_{i=1}^n A[i, j]$

B. $\sum_{j=1}^n A[i, j]$

C. $\sum_{i=1}^n A[j, i]$

D. $\sum_{j=1}^n A[i, j] + \sum_{j=1}^n A[j, i]$

(9) 若将无向图 $G = (V, E), |V| = n$ 用邻接表表示, 则在该邻接表中最多有 () 个边表结点。

A. n^2

B. $n(n-1)$

C. $n(n+1)$

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

(10) 在有向图的邻接表存储结构中, 顶点 v 在链表 (即边表) 中出现的次数为 ()。

- A. 顶点 v 的度 B. 顶点 v 的出度 C. 顶点 v 的入度 D. 依赖顶点 v 的边数

(11) 下面说法正确的是()。

- A. 邻接多重表是有向图和无向图的存储结构
- B. 十字链表是有向图和无向图的存储结构
- C. 邻接多重表是无向图的存储结构,十字链表是有向图的存储结构
- D. 邻接多重表是有向图的存储结构,十字链表是无向图的存储结构

(12) 用邻接表存储图的所用的空间大小为()。

- A. 与图的顶点数和边数都有关
- B. 只与图的边数有关
- C. 只与图的顶点数有关
- D. 与边数的平方有关

(13) 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 ()。

- A. $N^2 - 3N$ B. $(N^2 - 2N)/2$ C. $(N^2 - 2N)/3$ D. $(N^2 - 2N)/6$

(14) 用邻接矩阵存储有 n 个顶点和 e 条边的有向图。删除结点 i 的时间复杂度是 ()。

- A. $O(1)$ B. $O(n)$ C. $O(e)$ D. $O(n+e)$

(15) 有向图的一顶点的入度为 k_1 、出度为 k_2 ，则逆邻接表中该顶点单链表中的结点数为 ()。

A. k_2

B. k_1

C. $k_1 - k_2$

D. $k_1 + k_2$

6.2.7 图的存储-真题演练(答案见原书页码 P207)

(20) 【2013】 设图的邻接矩阵 $A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$, 各顶点的度依次是 ()。

A. 1,2,1,2

B. 2,2,1,1

C. 3,4,2,3

D. 4,4,2,2

6.3.3 图的遍历-习题精编(答案见原书页码 P218)

(1) 从无向图的任意顶点出发进行一次深度优先搜索便可访问所有顶点,则该图一定是()。

- A. 完全图 B. 连通图 C. 有回路 D. 一棵树

(2) 以下说法, 正确的是 ()。

*I.*用邻接表存储的图的深度优先遍历类似于树的先序遍历; 而用邻接矩阵存储的图的广度优先遍历类似于树的层次遍历

*II.*用邻接矩阵存储的图的深度优先遍历类似于树的先序遍历; 而用邻接表存储的图的广度优先遍历类似于树的层次遍历

*III.*从连通图中任一顶点出发进行图的遍历 (包含 *DFS* 和 *BFS*) 一定可访问到所有顶点

*IV.*对于一张非连通图而言, 对该图进行深度优先遍历, 需调用 *DFS* 函数 (以教辅所给的函数而言) 的次数刚好是图的连通分量数

*V.*深度优先遍历和广度优先遍历既适用于无向图, 也适用于有向图; 既适用于无权图, 也适用于有权图

A. 除 *IV* 以外皆正确 B. *I*、*II*、*V* C. 除 *V* 以外皆正确 D. 全部正确

(3) 在下面序列中,满足右图的深度优先遍历规则的序列是()。

① *afbdecg*

② *acfdebg*

③ *afbcdeg*

④ *agefdbc*

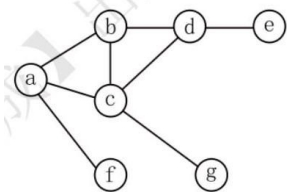
⑤ *afbcgde*

A. 1,3,5

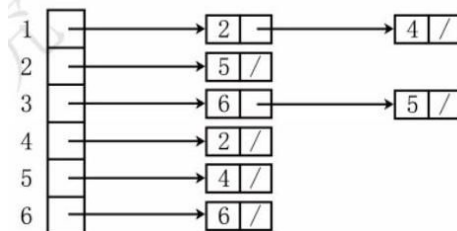
B. 1,3,4

C. 全部

D. 1,2,3



(4) 一个有向图 G 用邻接表表示如右图所示,以顶点 1 为起始点,以下满足深度优先遍历规则的序列是 ()。



A. 125436

B. 124536

C. 124563

D. 362514

(5) 无向图 $G = (V, E)$, 其中 $V = \{a, b, c, d, e, f\}$, $E = \{(a, b), (a, e), (a, c), (b, e), (c, f), (f, d), (e, d)\}$, 以 a 为源点, 对其进行深度优先遍历, 得到顶点序列正确的是 ()。

A. a, b, e, c, d, f

B. a, c, f, e, b, d

C. a, e, b, c, f, d

D. a, e, d, f, c, b

(6) 如右图所示,在下面的 5 个序列中,符合广度优先遍历的序列有 () 个。

① *abcfdge*

② *acfgdeb*

③ *afcbgde*

④ *agbecdf*

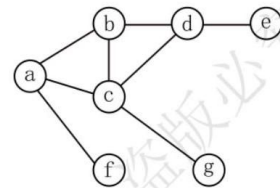
⑤ *acbefdg*

A. 5

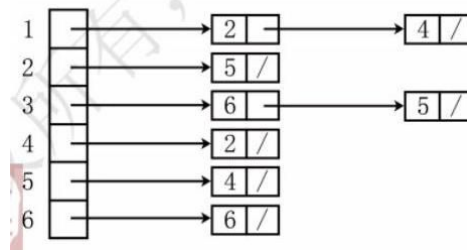
B. 4

C. 3

D. 2



(7) 有向图 G 用邻接表表示如右图所示, 若对该图按照广度优先算法从顶点 1 出发开始遍历 (不一定按照该邻接表遍历), 下列说法正确的是 ()。



- A. 图 G 的强连通分量数为 2
- B. 广度优先遍历的一个序列是 142536
- C. 广度优先遍历的所有序列是 124536
- D. 广度优先遍历的一个序列是 125436

(8) 无向图 $G = (V, E)$, 其中: $V = \{a, b, c, d, e, f\}$, $E = \{(a, b), (a, e), (a, c), (b, e), (c, f), (f, d), (e, d)\}$, 以顶点 a 为源点, 对该图进行广度优先遍历, 得到的顶点序列正确的是 ()。

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

(9) 现有图 G , 含有 n 个顶点、 e 条边。若使用邻接矩阵存储图 G , 则进行 DFS 遍历所需的时间复杂度为 (), 进行 BFS 遍历所需的时间复杂度为 (); 若使用邻接表存储图 G , 则进行 DFS 遍历所需的时间复杂度为 (), 空间复杂度为 (), 进行 BFS 遍历所需的时间复杂度为 (), 空间复杂度为 ()。

A. $O(n)$

B. $O(e)$

C. $O(n+e)$

D. $O(n^2)$

(10) 执行 () 操作时, 需要使用队列作为辅助空间。

- A. 查找哈希表
- B. 广度优先搜索图
- C. 先序(根)遍历二叉树
- D. 深度优先搜索图

6.3.4 图的遍历-真题演练(答案见原书页码 P218)

(16) 【2012】对有 n 个结点、 e 条边且使用邻接表存储的有向图进行广度优先遍历, 其算法时间复杂度是 ()。

A. $O(n)$

B. $O(e)$

C. $O(n+e)$

D. $O(n*e)$

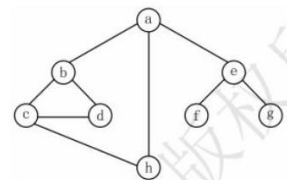
(17) 【2013】 若对右图所示的无向图进行遍历,则下列选项中,不是广度优先遍历序列的是()。

A. h, c, a, b, d, e, g, f

B. e, a, f, g, b, h, c, d

C. d, b, c, a, h, e, f, g

D. a, b, c, d, h, e, f, g



(18) 【2015】设有向图 $G=(V,E)$, 顶点集 $V=\{v_0, v_1, v_2, v_3\}$, 边集 $E=\{<v_0, v_1>, <v_0, v_2>, <v_0, v_3>, <v_1, v_3>\}$ 。若从顶点 v_0 开始对图进行深度优先遍历, 则可得遍历序列个数是()。

A. 2

B. 3

C. 4

D. 5

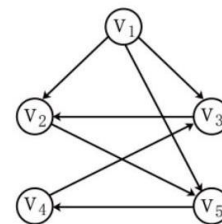
(19) 【2016】下列选项中, 不是右图深度优先搜索序列的是 ()。

A. V_1, V_5, V_4, V_3, V_2

B. V_1, V_3, V_2, V_5, V_4

C. V_1, V_2, V_5, V_4, V_3

D. V_1, V_2, V_3, V_4, V_5



6.4.5 图的基本应用-习题精编(答案见原书页码 P250)

(1) 用 DFS 遍历一个无环有向图,并在 DFS 算法退栈返回时打印相应的顶点,则输出的顶点序列是 ()。

- A. 逆拓扑有序 B. 拓扑有序 C. 无序的 D. 按关键字有序

(2) 有关拓扑排序的表述, 下列说法中不正确的是()。

I. 若图的拓扑排序唯一, 那么该拓扑排序对应的图也唯一

II. 拓扑排序算法实现时, 栈和队列都可以作为保存入度为零的顶点信息的数据结构

III. 若某有向图的拓扑有序序列唯一, 那么该图中每个顶点的入度最多为 1, 出度可以不是 1

A. *I*、*III*

B. *II*、*III*

C. *II*

D. *III*

(3) 在有向图 G 的拓扑序列中, 若顶点 v_i 在顶点 v_j 之前, 则下列情形不可能出现的是 ()。

- A. G 中有弧 $\langle v_i, v_j \rangle$
- B. G 中有一条从 v_i 到 v_j 的路径
- C. G 中没有弧 v_i, v_j
- D. G 中有一条从 v_j 到 v_i 的路径

(4)(多选题)下面哪些方法可以判断一个有向图是否有环(回路)()。

- A. 深度优先遍历
- B. 拓扑排序
- C. 求最短路径
- D. 求关键路径(涉及拓扑序列算法)

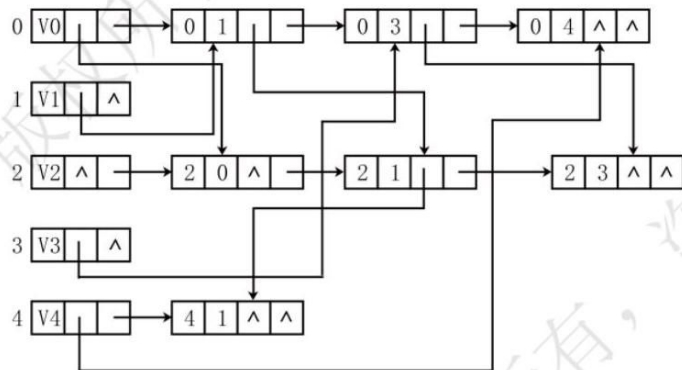
(5)(多选题) 若一个有向图的顶点不能排在一个拓扑序列中, 则可判定该有向图 ()。

- A. 是一个有根的有向图
- B. 是一个强连通图
- C. 含有多个入度为 0 的顶点
- D. 含有多个出度为 0 的顶点
- E. 含有顶点数大于 1 的强连通分量
- F. 有向图必有环

(6) 如果一个有向图具有拓扑序列,那么该图的邻接矩阵一定是();如果一个有向图具有有序的拓扑序列,那么该图的邻接矩阵一定是()。

- A. 对称矩阵 B. 稀疏矩阵 C. 三角矩阵 D. 普通的矩阵

(7) 图 G 利用十字链表法表示如下, 请问图 G 可能的拓扑序列为 ()。



- 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

(8) 以下与图有关的表述中, 正确的是 ()。

*I.*邻接矩阵中第 V 行中的 1 的个数就等于有向图中顶点 V 的度

*II.*求最小生成树时,权值为负可能会对结果产生影响

*III.*在图 G 的最小生成树 G' 中, 某条边的权值有可能会大于某条未入选边的权值

*IV.*如果有向无环图 (DAG) 的拓扑序列是唯一确定的,那么该 DAG 也是唯一确定的

A. *I*、*II* 和 *III*

B. *III* 和 *IV*

C. *III*

D. *IV*

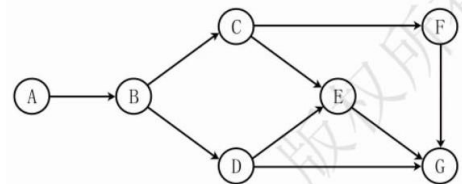
(9) 如右图所示有向图的拓扑序列共有 () 个；若将右图的所有有向边反向,则此时该有向图的拓扑序列有 () 个 (即逆拓扑序列的个数)。

A. 4

B. 6

C. 5

D. 7



(10) 已知有向图 $G = (V, E)$, 其中 $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$, $E = \{<v_1, v_2>, <v_1, v_3>, <v_1, v_4>, <v_2, v_5>, <v_3, v_5>, <v_3, v_6>, <v_4, v_6>, <v_5, v_7>, <v_6, v_7>\}$, G 的拓扑序列是 ()。

A. $v_1, v_3, v_4, v_6, v_2, v_5, v_7$

B. $v_1, v_3, v_2, v_6, v_4, v_5, v_7$

C. $v_1, v_3, v_4, v_5, v_2, v_6, v_7$

D. $v_1, v_2, v_5, v_3, v_4, v_6, v_7$

(11) 以下对于最小生成树的描述,正确的是()。

I.所有无向连通图的最小生成树一定有多个

II. *Prim* 和 *Kruskal* 算法构造出的最小生成树一定相同

III.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

IV.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

V.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 的子图就是图 G 的生成树

VI.生成树就是最小生成树

A. III

B. III、IV

C. 全部正确

D. 全部错误

(12) 对于含有 n 个顶点的带权连通图, 其最小生成树是指图中任一 ()。

- A. 由 $n-1$ 条权值最小的边构成的子图
- B. 由 $n-1$ 条权值之和最小的边构成的子图
- C. 由 $n-1$ 条权值之和最小的边构成的连通子图
- D. 由 n 个顶点构成的边的权值之和最小的连通子图

(13) 若一带权的连通无向图中 $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$, $E = \{(v_1, v_2), (v_1, v_3), (v_3, v_4), (v_3, v_6), (v_2, v_5), (v_4, v_5), (v_4, v_6), (v_5, v_7), (v_6, v_7)\}$, 其边对应的权值分别是 $\{10, 2, 2, 11, 1, 4, 6, 7, 3\}$, 请问从 v_1 到 v_7 的最短路径所经过的顶点序列 (包含 v_1 和 v_7) 的第 3 个顶点和最短路径长度分别为 ()。

A. $v_5; 18$

B. $v_4; 13$

C. $v_4; 15$

D. $v_5; 24$

(14) 以下的说法正确的是 ()。

I. 求解 *Floyd* 算法的时间复杂度为 $O(n^3)$

II. 当各条边上的权值均相等时, *BFS* 算法可以用来求解单源最短路径问题

III. 最短路径一定是简单路径

IV. *Dijkstra* 算法不适用于含有边权为负的图求解最短路径问题, 且要求所求的图不能有回路。该算法一般用于求解单源最短路径问题

V. *Floyd* 算法不适用于含有负权圈的图求解最短路径问题。该算法一般用于求解多源最短路径问题

VI. *Floyd* 算法求解两个顶点的最短路径时, $path^{k-1}$ 一定是 $path^k$ 的子集

A. I、II、III、V

B. I、II、III

C. 全部正确

D. 全部错误

(15) 若某带权图为 $G = (V, E)$, $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$, $E = \{<v_1, v_2>5, <v_1, v_3>6, <v_2, v_5>3, <v_3, v_5>6, <v_3, v_4>3, <v_4, v_5>3, <v_4, v_7>1, <v_4, v_8>4, <v_5, v_6>4, <v_5, v_7>2, <v_6, v_{10}>4, <v_7, v_9>5, <v_8, v_9>2, <v_9, v_{10}>2\}$ (注: 边括号外的数据表示边上的权值), 则 G 的关键路径的长度为 ()。

A. 19

B. 20

C. 21

D. 22

(16) 下面关于关键路径的说法,错误的是()。

- A. 求关键路径是以拓扑排序为基础的
- B. 一个事件的最早发生时间与以该事件为始的弧的活动的最早开始时间相同
- C. 一个事件的最迟发生时间是以该事件为尾的弧的活动的最迟开始时间与该活动的持续时间之差
- D. 关键活动一定位于关键路径上

(17) 下列关于关键路径的说法中, 正确的是()。

*I.*如果变更了 AOE 网的某条关键路径上的任意一个关键活动,则关键路径一定会发生变化,并产生新的关键路径

*II.*关键路径上关键活动的花销增加了多少,整个工程的花销也一定会增加多少;增加非关键路径的花销也可能影响最终关键路径的花销

*III.*如果减少关键路径上任一关键活动的花销,则关键路径的长度(花销)会减少;但减少非关键路径的花销一定不会影响关键路径的花销

*IV.*若减少所有关键路径上共有的任一关键活动的花销,则关键路径的长度(花销)会减少

*V.*只有增加所有关键路径上共有的关键活动的花销,关键路径的长度(花销)会增加;增加非关键路径的花销不会影响最终的关键路径的长度(花销)

A. *II* 和 *V*

B. *I*、*II* 和 *IV*

C. *II* 和 *IV*

D. *I* 和 *IV*

(18) 关键路径是 AOE 网中 ()。

- A. 从起始点到终点的最短路径
- B. 从起始点到终点的最长路径
- C. 从起始点到终点的边数最多的路径
- D. 从起始点到终点的边数最少的路径

(19) 下列关于 *AOE* 网的描述中,错误的是 ()。

- 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. 某些关键活动若提前完成, 那么整个工程将会提前完成

(20) 在求边稠密的图的最小生成树时,采用什么算法会比较合适 ()。

A. *Prim*

B. *Kruskal*

C. *Dijkstra*

D. 其他

(21) 以下说法错误的是()。

I.求从指定源点到其余各顶点的迪杰斯特拉 (*Dijkstra*) 最短路径算法中弧上权不能为负的原因是在实际应用中无意义

II.若图用邻接矩阵表示,则利用 *Dijkstra* 算法求每一对不同顶点之间的最短路径的算法时间是 $O(n^3)$

III. *Floyd* 算法求每对不同顶点对的算法中允许弧上的权为负,但不能有权和为负的回路

A. I、II、III

B. I

C. I、III

D. II、III

(22) 求解有向图的关键路径时, 若该有向图用邻接矩阵表示且第 i 列值全部为无穷, 则 ()。

- A. 若关键路径存在, 第 i 个顶点一定是起点
- B. 若关键路径存在, 第 i 个顶点一定是终点
- C. 关键路径不存在
- D. 该有向图对应的无向图存在多个连通分量

(23) 下列各选项能用于求无向图的所有连通分量的是()。

- A. *BFS* B. 拓扑排序 C. 求最短路径 D. 求关键路径

(24) 带权连通图 $G = (V, E)$, 其中 $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_3), (v_2, v_4), (v_2, v_5), (v_3, v_4), (v_4, v_5)\}$ (注: 边的权值为顶点对中下标较大的数字), G 的最小生成树的权值之和为 ()。

A. 11

B. 12

C. 14

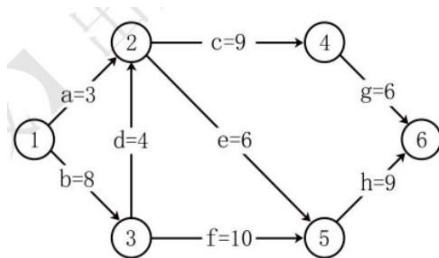
D. 15

(25) 在一个带权图 G 中权值最小的边一定包含在 G 的 ()。

- A. 深度优先生成树中
- B. 某棵最小生成树中
- C. 广度优先生成树中
- D. 任一最小生成树中

(26) 右图 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,加快进度就可以缩短工期的是 ()。

- A. c 和 e
- B. d 和 c
- C. f 和 d
- D. f 和 h

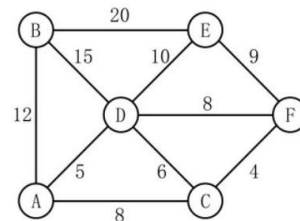


(27) 存在一张无向连通图 $G = (V, E)$, $|V| = n$, $|E| = e$ 。分别使用 *Prim* 算法和 *Kruskal* 算法来产生图 G 的最小生成树, 则时间复杂度分别是 ()。

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

(28) 存在一张无向图 G 如右图所示,现使用 **Prim** 算法和 **Kruskal** 算法来求解图 G 的最小生成树,则使用两个算法每次产生的边顺序相同的个数为 () (假设 **Prim** 算法以顶点 A 为起点)。

- A. 1
- B. 2
- C. 3
- D. 0D. $O(n^3)$



(29) 在图采用邻接表存储时,求最小生成树的 *Prim* 算法时间复杂度为 (假设图有 n 个顶点, e 条边) ()。

- A. $O(n)$ B. $O(n+e)$ C. $O(n^2)$ D. $O(n^3)$

(30) (选做题) 图 G 含有 6 个顶点, 18 条边, 则利用 *Dijkstra* 算法求解图 G 的单源最短路径时, 使用 () 结构保存候选最短路径耗费, 性能最好。

- A. 无序线性表 B. 有序线性表 C. 最小堆 D. 二叉搜索树

(31) 已知 7 个城市 (分别编号为 0~6) 之间修建道路的耗费分别为:0~1:22,0~2:9,0~3:10,1~3:15,1~4:7,1~6:12,2~3:4,2~5:3,3~5:5,3~6:23,4~6:20,5~6:32, 要修建路网让每两个城市之间都可以互通 (直达或经过其他城市), 最小的耗费是 ()。

A. 50

B. 55

C. 35

D. 38

(32) 迪杰斯特拉 (*Dijkstra*) 算法的基本思想是 ()。

- A. 按路径长度递减的次序产生最短路径
- B. 按路径长度递增的次序产生最短路径
- C. 按广度优先遍历的次序产生最短路径
- D. 按深度优先遍历的次序产生最短路径

(33) 求解单源最短路径的 *Dijkstra* 算法和所有顶点对最短路径的 *Floyd – Warshall* 算法分别使用了设计算法的哪两种技术()。

- A. 贪心、动态规划 B. 动态规划、贪心 C. 贪心、贪心 D. 动态规划、动态规划

(34) 下列哪种图的邻接矩阵是对称矩阵？（ ）

A. 有向图

B. 无向图

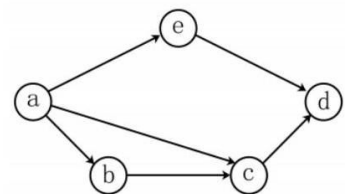
C. *AOV*网

D. *AOE*网

6.4.6 图的基本应用-真题演练(答案见原书页码 P250)

(36) 【2010】对右图进行拓扑排序, 可得不同拓扑排序的个数是 ()。

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



(37) 【2011】下列关于图的叙述中,正确的是()。

*I.*回路是简单路径

*II.*存储稀疏图,用邻接矩阵比邻接表更省空间

*III.*若有向图中存在拓扑序列,则该图不存在回路

A. 仅 *II*

B. 仅 *I*、*II*

C. 仅 *III*

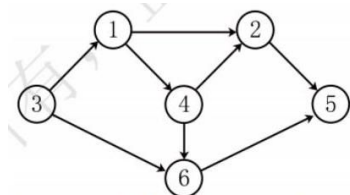
D. 仅 *I*、*III*

(38) 【2012】若用邻接矩阵存储有向图, 矩阵中主对角线以下的元素均为零, 则关于该图的拓扑序列的结论是()。

- A. 存在, 且唯一 B. 存在, 且不唯一 C. 存在, 可能不唯一 D. 无法确定是否存在

(39) 【2014】对右图所示的有向图进行拓扑排序,得到的拓扑序列可能是()。

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



(40) 【2016】若 n 个顶点、 e 条弧的有向图用邻接表存储, 则拓扑排序算法时间复杂度是 ()。

A. $O(n)$

B. $O(n+e)$

C. $O(n^2)$

D. $O(ne)$

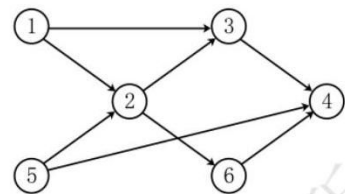
(41) 【2018】下列选项中不是右侧有向图的拓扑序列的是 ()。

A. 1, 5, 2, 3, 6, 4

B. 5, 1, 2, 6, 3, 4

C. 5, 1, 2, 3, 6, 4

D. 5, 2, 1, 6, 3, 4



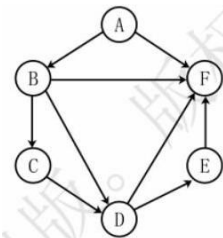
(42) 【2021】 给定右侧有向图,该图的拓扑有序序列的个数是 ()。

A. 1

B. 2

C. 3

D. 4



(43) 【2012】下列关于最小生成树的陈述中, 正确的是()。

*I.*最小生成树的代价唯一

*II.*所有权值最小的边一定会出现在所有的最小生成树中

*III.*使用 **Prim** 算法从不同顶点开始得到的最小生成树一定相同

*IV.*使用 **Prim** 算法和 **Kruskal** 算法得到的最小生成树总不相同

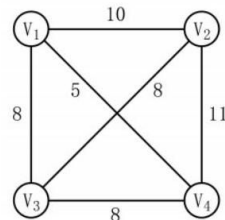
A. 仅 *I*

B. 仅 *II*

C. 仅 *I*、*III*

D. 仅 *II*、*IV*

(44) 【2015】求右侧带权图的最小(代价)生成树时,可能是 *Kruskal* 算法第 2 次选中但不是 *Prim* 算法(从 V_4 开始)第 2 次选中的边是()。



A. (V_1, V_3)

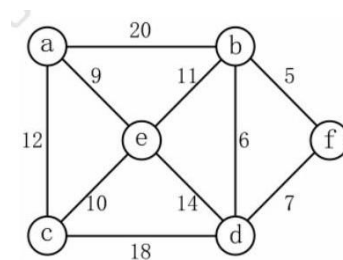
B. (V_1, V_4)

C. (V_2, V_3)

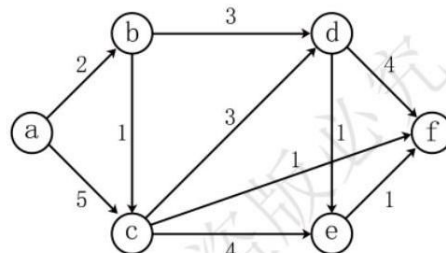
D. (V_3, V_4)

(45) 【2020】已知无向图 G 如右所示, 使用克鲁斯卡尔 (*Kruskal*) 算法求图 G 的最小生成树, 加到最小生成树中的边依次是 ()。

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



(46) 【2012】对右图所示的有向带权图,若采用 *Dijkstra* 算法求从源点 a 到其他各顶点的最短路径,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依次是 ()。



- A. d, e, f
- B. e, d, f
- C. f, d, e
- D. f, e, d

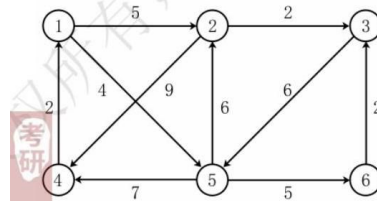
(47) 【2016】使用 *Dijkstra* 算法求右图中从顶点 1 到其他各顶点的最短路径,依次得到的各最短路径的目标顶点是()。

A. 5,2,3,4,6

B. 5,2,3,6,4

C. 5,2,4,3,6

D. 5,2,6,3,4



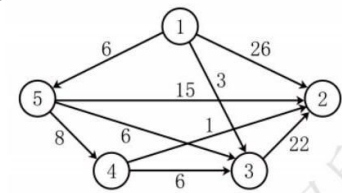
(48) 【2021】使用 *Dijkstra* 算法求右图中从顶点 1 到其余各顶点的最短路径,将当前找到的从顶点 1 到顶点 2,3,4,5 的最短路径长度保存在数组 *dist* 中,求出第二条最短路径后, *dist* 内容更新为 ()。

A. 26,3,14,6

B. 25,3,14,6

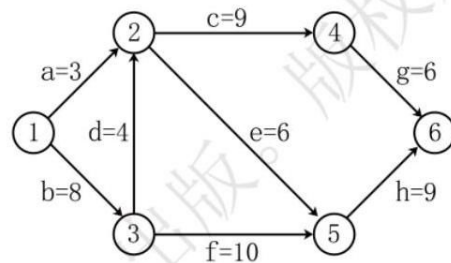
C. 21,3,14,6

D. 15,3,14,6



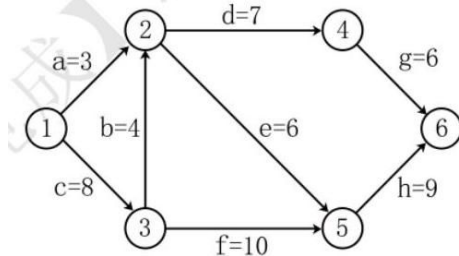
(49) 【2013】右侧的 AOE 网表示一项包含 8 个活动的工程。通过同时加快若干活动的进度可以缩短整个工程的工期。下列选项中, 加快其进度就可以缩短工期工程的是 ()。

- A. c 和 e
- B. d 和 c
- C. f 和 d
- D. f 和 h



(50) 【2019】如右图所示的 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 和 7
- B. 12 和 12
- C. 12 和 14
- D. 15 和 15



(51) 【2020】若使用 AOE 网估算工程进度, 则下列叙述中正确的是()。

- A. 关键路径是从源点到汇点边数最多的一条路径
- B. 关键路径是从源点到汇点路径长度最长的路径
- C. 增加任一关键活动的时间不会延长工程的工期
- D. 缩短任一关键活动的时间将会缩短工程的工期

(52) 【2019】用有向无环图描述表达式 $(x+y) * ((x+y)/x)$, 需要的顶点个数至少是 ()。

A. 5

B. 6

C. 8

D. 9

(53) 【2020】修改递归方式实现的图的深度优先搜索 (DFS) 算法, 将输出 (访问) 顶点信息的语句移到退出递归前 (即执行输出语句后立刻退出递归)。采用修改后的算法遍历有向无环图 G , 若输出结果中包含 G 中的全部顶点, 则输出的顶点序列是 G 的 ()。

- A. 拓扑有序序列
- B. 逆拓扑有序序列
- C. 广度优先搜索序列
- D. 深度优先搜索序列

第 7 章查找

7.2.4 顺序查找、折半查找与分块查找-习题精编(答案见原书页码 P279)

(1) 顺序查找法适合于存储结构为()的线性表。

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

(2) 设线性表的长度为 n , 则顺序查找成功的平均比较次数为 ()。

A. n

B. $n/2$

C. $(n+1)/2$

D. $(n-1)/2$

(3) 查找长度为 3 的顺序表, 若对第 1 个元素的查找概率为 $1/12$, 对第 2 个元素的概率为 $2/3$, 对第 3 个元素的概率为 $1/4$, 则查找任一元素的平均查找长度为 ()。

A. $13/6$

B. 2

C. $7/3$

D. $4/3$

(4) 假设顺序表中包含 5 个数据元素 (a, b, c, d, e) , 它们的查找概率分别为 $\{0.3, 0.35, 0.2, 0.1, 0.05\}$, 在顺序查找时, 为了使查找成功的平均查找长度最小, 则表中数据元素的存放顺序是 ()。

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

(5) 使用折半查找法对有序表 (9,15,34,44,49,61,73,85,99,102,112) 查找值为 99 的元素时, 查找成功的比较次数为()。

A. 1

B. 2

C. 4

D. 6

(6) 当在一个有序的顺序表上查找一个数据时,既可以用折半查找法,也可以使用顺序查找法,但前者的查找速度肯定比后者的查找速度()。

- A. 必然快
- B. 必然慢
- C. 取决于表是递增还是递减
- D. 不一定

(7) 在顺序有序表中 (2,5,7,10,14,15,18,23,35,41,52), 用折半查找法查找关键字 14 的关键字比较次数为()。用折半查找法查找关键字 6 的关键字比较次数为()。

A. 2

B. 3

C. 4

D. 5

(8) 对一个长度为 50 的有序表进行折半查找,最多比较 () 次就能查找出结果。

A. 6

B. 7

C. 8

D. 9

(9) 折半查找有序表 (2, 10, 25, 35, 40, 65, 70, 75, 81, 82, 88, 100), 若查找元素 75, 需要依次与表中元素 () 进行比较。

A. 65, 82, 75

B. 70, 82, 75

C. 65, 81, 75

D. 65, 81, 70, 75

(10) 对线性表进行二分查找时,要求线性表必须()。

- A. 以顺序方式存储, 且数据元素无需有序
- B. 以顺序方式存储, 且数据元素有序
- C. 以链接方式存储, 且数据元素无需有序
- D. 以链接方式存储, 且数据元素有序

(11) 当采用分块查找时, 数据的组织方式为()。

- A. 数据分成若干块,每块内数据有序
- B. 数据分成若干块,每块内数据不必有序,但块间必须有序,每块内最大(或最小)的数据组成索引块
- C. 数据分成若干块,每块内数据有序,每块内最大(或最小)的数据组成索引块
- D. 数据分成若干块,每块(除最后一块外)中数据个数需要相同

(12) 在下列查找算法中,平均查找速度最快的是()。

- A. 顺序查找 B. 折半查找 C. 分块查找 D. 二叉排序树查找

(13) 既希望较快地查找又便于线性表动态变化的查找算法是()。

- A. 顺序查找 B. 折半查找 C. 索引顺序查找 D. 哈希法查找

(14) 下列说法中, 正确的是 ()。

- A. 如果数据元素保持有序, 则查找时就可以采用折半查找法
- B. 折半查找与二叉查找树的时间性能在最坏的情况下是相同的
- C. 折半查找法的查找速度一定比顺序查找法快
- D. 折半查找法查找一个元素平均需要 $\log_2 n$ 次关键字比较

7.2.5 顺序查找、折半查找与分块查找-真题演练(答案见原书页码 P279)

(15) 【2010】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的是 ()。

A. 4

B. 5

C. 6

D. 7

(16) 【2015】下列选项中不能构成折半查找中关键字比较序列的是()。

- A. 500, 200, 450, 180 B. 500, 450, 200, 180 C. 180, 500, 200, 450 D. 180, 200, 500, 450

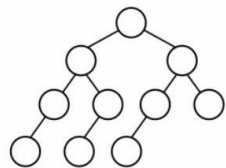
(17) 【2016】在有 $n(n > 1000)$ 个元素的升序数组 A 中查找关键字 x 。查找算法的伪代码如下所示：

```
k=0;
while(k<n && A[k]<x) k=k+3;
if(k<n && A[k]==x) 查找成功;
elseif(k-1<n && A[k-1]==x) 查找成功;
elseif(k-2<n && A[k-2]==x) 查找成功;
else 查找失败;
```

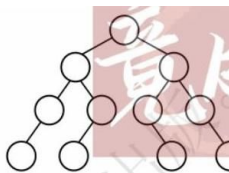
本算法与折半查找算法相比，有可能具有更少比较次数的情形是（ ）。

- A. 当 x 不在数组中
- B. 当 x 接近数组开头处
- C. 当 x 接近数组结尾处
- D. 当 x 位于数组中间位置

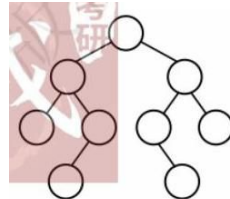
(18) 【2017】以下二叉树中,可能成为折半查找判定树(不含外部结点)的是()。



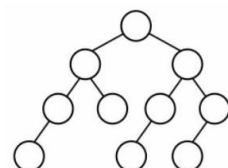
A



B



C



D

7.3.4 二叉搜索树、平衡二叉树-习题精编(答案见原书页码 P301)

(1) 下列说法中正确的是 ()。

- A. 任何一棵含 n 个结点的二叉查找树, 可以通过 $O(n)$ 次旋转, 转变为另一棵含 n 个结点的二叉查找树
- B. 满足任一分支结点的值都小于其右孩子的值, 大于其左孩子的值的二叉树就是二叉查找树
- C. 假设在一颗 BST 中查找一个关键字 k , 查找结束于一个叶结点, 设 A 集合为查找路径左侧关键字的集合; B 是查找路径上的集合; C 是查找路径右侧关键字的集合。则 $\forall a \in A, b \in B, c \in C$, 满足 $a \leq b \leq c$ 。
- D. 一个序列仅能构成一种 AVL 树

(2) 二叉排序树中, 按()遍历二叉排序树得到的序列是一个有序序列。

A. 先序

B. 中序

C. 后序

D. 层次

(3) 一棵具有 N 个结点的二叉排序树, 查找某个关键字的结点时, 最多进行 () 次比较, 最少进行 () 次比较, 理想状态下查找叶子结点最多需要比较 () 次。

A. $\lfloor \log_2(N+1) \rfloor$

B. 1

C. N

D. $\lceil \log_2(N+1) \rceil$

(4) 二叉排序树中最大的结点 ()。

- A. 没有左孩子
- B. 没有右孩子
- C. 既有左孩子, 又有右孩子
- D. 没有左孩子, 也没有右孩子

(5) 对于二叉排序树, 下面的说法中正确的是()。

- A. 对二叉排序树进行先序遍历可得到有序序列
- B. 用逐点插入法构造二叉排序树, 若先后插入的关键字有序, 二叉排序树的深度最大
- C. 在二叉排序树中进行查找, 关键字的比较次数不超过结点数的 $1/2$
- D. 二叉排序树是动态树表, 查找失败时插入新结点, 会引起树的重新分裂和组合

(6) 完全二叉排序树的平均查找时间复杂度为 ()。

A. $O(\log_2 n)$

B. $O(n)$

C. $O(1)$

D. $O(n^2)$

(7) 若想查找 63, 哪个最有可能是在二叉排序树上进行的查找 ()。

A. 2, 25, 101, 39, 80, 70, 59, 63

B. 39, 101, 25, 80, 70, 59, 63

C. 101, 70, 2, 39, 25, 59, 80, 63

D. 59, 2, 80, 70, 39, 25, 63

(8) 在关键字随机分布的情况下,用二叉排序树方法进行查找,下列方法中与其平均查找长度数量级相当的是()。

- A. 顺序查找 B. 折半查找 C. 分块查找 D. 均不正确

(9) 如果按照以下列序列构造二叉排序树,选项()与其他3个序列构造的结果不同。

A. (98,83,90,61,120,115,130)

B. (98,120,115,130,83,61,90)

C. (98,61,83,90,120,115,130)

D. (98,83,61,90,120,130,115)

(10) 按照序列 (45, 23, 16, 33, 70, 72, 94, 58, 25, 47) 构造一颗二叉排序树, 在这棵树中查找 47 需要比较 () 次。

A. 3

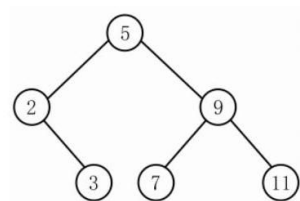
B. 4

C. 5

D. 6

(11) 对图 7.22 所示的二叉排序树进行遍历 (层次遍历, 先序遍历, 中序遍历或者后序遍历), 不能生成以下哪个序列 ()。

- A. (5,2,9,3,7,11)
- B. (2,3,5,7,9,11)
- C. (5,2,3,9,7,11)
- D. (3,2,5,7,9,11)



(12) 分别按照下列序列构造二叉排序树, 不能得到如图 7.22 所示的二叉排序树的选项是 ()。

A. 5, 7, 9, 11, 2, 3

B. 5, 2, 3, 9, 11, 7

C. 5, 2, 9, 7, 11, 3

D. 5, 9, 7, 11, 2, 3

(13) 如图 7.22 所示的二叉排序树,查找成功时的平均查找长度为(),查找失败时的平均查找长度为()。

A. 3,6

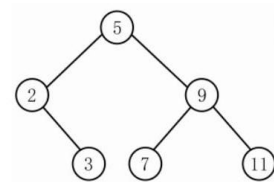
B. 13/6,20/7

C. 7/3,20/7

D. 3,10/3

(14) 如图 7.22 所示的二叉排序树, 删除关键字为 5 的结点后, 根结点的右孩子结点为 ()。

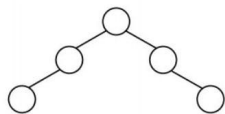
- A. 3
- B. 7
- C. 9
- D. 11



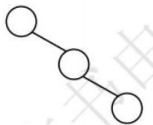
(15) 对相同的 n 个整数构成的二叉排序树和小根堆, 下面哪个说法是不正确的 ()。

- A. 二叉排序树高度大于等于小根堆高度
- B. 对该二叉排序树进行中序遍历可得到从小到大的序列
- C. 从小根堆根结点到其任何叶结点的路径上的结点值构成从小到大的序列
- D. 对该小根堆进行按层次 (*levelorder*) 遍历可得到从小到大的序列

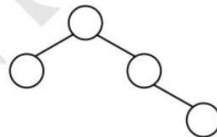
(17) 下列二叉树中, 不满足平衡二叉树的定义是 ()。



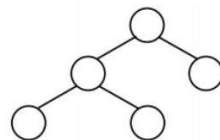
A



B



C



D

(18) 若平衡二叉树 (AVL 树) 的结点数是 21, 则该树的高度至多是多少?(只有一个根结点的树高度为 0) ()。

A. 4

B. 5

C. 6

D. 7

(19) 在具有 12 个结点的平衡二叉树 (AVL 树) 中,查找一个关键字最多需要 () 次比较。

A. 3

B. 4

C. 5

D. 6

(20) 在 *AVL* 树中,() 可能会发生两次旋转调整。

- A. 添加、删除结点操作
- B. 仅删除结点操作
- C. 仅添加结点操作
- D. 添加、删除结点都不

(21) 将 1、2、3、4、5、6 顺序插入初始为空的平衡二叉树 (*AVL*) 树中, 当完成这 6 个元素的插入后, 该 *AVL* 树共有 () 层 (结点的层数)。

A. 2

B. 3

C. 4

D. 5

(22) 将关键字 $1, 2, 3, \dots, 2016$ 插入初始为空的平衡二叉树中, 假设只有一个根结点的二叉树高度为 0, 那么最终二叉树的高度为 ()。

A. 7

B. 8

C. 9

D. 10

(23) 将数字序列 {40, 54, 75, 96, 63, 69, 30} 插入 AVL 树中, 下列说法错误的是()。

- A. 该树是完全二叉树
- B. 63 为根结点
- C. 30 为叶结点
- D. 54 和 69 是兄弟结点

(24) 对具有 15 个结点的平衡二叉树进行查找关键字 32 的操作, 则依次比较的关键字可能是下面选项中的 ()。

A. 36, 45

B. 34, 38, 22

C. 49, 11, 39, 32

D. 56, 30, 55, 41, 23, 32

(25) 在 *AVL* 树中插入一个结点后造成了不平衡, 设最低的不平衡结点为 *A*, 已知 *A* 的左孩子平衡因子为 0, 右孩子平衡因子为 1, 则应作 () 型调整使其平衡。

A. *LL*B. *LR*C. *RL*D. *RR*

(26) 一棵深度为 m 的平衡二叉树, 其每个结点的平衡因子均为 0, 则该树的结点数是 ()。

A. 2^m

B. $2^m - 1$

C. 2^{m-1}

D. 无法确定

(27) 设输入序列为 {20,15,35,16,17}, 构造一棵平衡二叉树, 当在树中插入值 17 时发生不平衡, 则应进行的平衡旋转是 ()。

A. *LL*

B. *RR*

C. *LR*

D. *RL*

(28) 具有 6 层结点的平衡二叉树 (AVL) 中至少有 () 个结点,至多有 () 个结点。

A. 20,63

B. 18,31

C. 33,64

D. 33,32

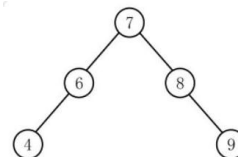
(29) 已知平衡二叉排序树 (简称平衡二叉树) 如右图所示,若插入关键字 3 后得到一棵新的平衡二叉树,则在新平衡二叉树中,关键字 4 所在结点的左右孩子结点保存的关键字分别是 ()。

A. 3,6

B. 6,7

C. 3,7

D. 6,3



7.3.5 二叉搜索树、平衡二叉树-真题演练(答案见原书页码 P301)

(33) 【2011】对于下列关键字序列, 不可能构成某二叉排序树中一条查找路径的序列是 ()。

A. 95, 22, 91, 24, 94, 71

B. 92, 20, 91, 34, 88, 35

C. 21, 89, 77, 29, 36, 38

D. 12, 25, 71, 68, 33, 34

(34) 【2013】在任意一棵非空二叉排序树 T_1 中, 删除某结点 v 之后形成二叉排序树 T_2 , 再将 v 插入 T_2 形成二叉排序树 T_3 。下列关于 T_1 与 T_3 的叙述中, 正确的是 ()。

I. 若 v 是 T_1 的叶结点, 则 T_1 与 T_3 不同 *II.* 若 v 是 T_1 的叶结点, 则 T_1 与 T_3 相同

III. 若 v 不是 T_1 的叶结点, 则 T_1 与 T_3 不同 *IV.* 若 v 不是 T_1 的叶结点, 则 T_1 与 T_3 相同

A. 仅 *I*、*III*

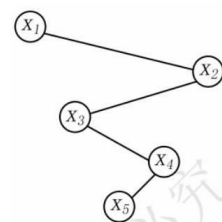
B. 仅 *I*、*IV*

C. 仅 *II*、*III*

D. 仅 *II*、*IV*

(35) 【2018】已知二叉排序树如右图所示, 元素之间应满足的大小关系是()。

- A. $X_1 < X_2 < X_5$
- B. $X_1 < X_4 < X_5$
- C. $X_3 < X_5 < X_4$
- D. $X_4 < X_3 < X_5$



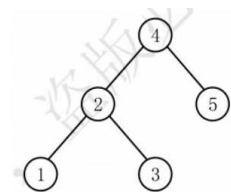
(36) 【2020】下列给定的关键字输入序列中, 不能输出如右图所示的二叉排序树的是 ()。

A. 4, 5, 2, 1, 3

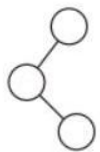
B. 4, 5, 1, 2, 3

C. 4, 2, 5, 3, 1

D. 4, 2, 1, 3, 5



(37) 【2009】 下图二叉排序树中, 满足平衡二叉树定义的是 ()。



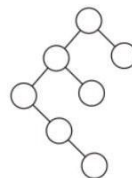
A



B



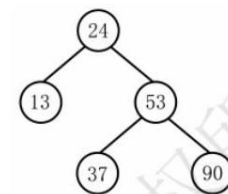
C



D

(38) 【2010】右图所示平衡二叉树中, 插入关键字 48 后得到一棵新平衡二叉树。在新平衡二叉树中, 关键字 37 所在结点的左、右子结点中保存的关键字分别是 ()。

- A. 13,48
- B. 24,48
- C. 24,53
- D. 24,90



(39) 【2012】若平衡二叉树的高度为 6, 且所有非叶结点的平衡因子均为 1, 则该平衡二叉树的结点总数为 ()。

A. 12

B. 20

C. 32

D. 33

(40) 【2013】若将关键字 1,2,3,4,5,6,7 依次插入初始为空的平衡二叉树 T 中, 则 T 中平衡因子为 0 的分支结点的个数是 ()。

A. 0

B. 1

C. 2

D. 3

(41) 【2015】现有一棵无重复关键字的平衡二叉树 (AVL 树), 对其进行中序遍历可得到一个降序序列。下列关于该平衡二叉树的叙述中, 正确的是 ()。

- A. 根结点的度一定为 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. 树中最大元素一定无左子树

(42) 【2019】在任意一棵非空平衡二叉树 (AVL 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 与 T_3 的叙述中, 正确的是 ()。

I. 若 v 是 T_1 的叶结点, 则 T_1 与 T_3 可能不相同

II. 若 v 不是 T_1 的叶结点, 则 T_1 与 T_3 一定不相同

III. 若 v 不是 T_1 的叶结点, 则 T_1 与 T_3 一定相同

A. 仅 I

B. 仅 II

C. 仅 I、II

D. 仅 I、III

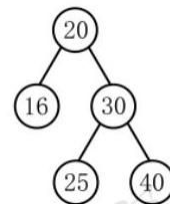
(43) 【2021】 给定平衡二叉树如右图所示,插入关键字 23 之后,根中的关键字是()。

A. 16

B. 20

C. 23

D. 25



7.4.6 B/B+树基础、红黑树-习题精编(答案见原书页码 P328)

(1) 下列关于 m 阶 B 树的说法中, 错误的是()。

- A. 根结点至多有 m 棵子树
- B. 所有叶结点都在同一层次上
- C. 非叶结点至少有 $m/2$ (m 为偶数) 或 $(m+1)/2$ (m 为奇数) 棵子树
- D. 根结点中的数据是有序的

(2) 在一棵具有 20 个关键字的 3 阶 B 树中,含关键字的结点个数最多是(),最少是()。

A. 10

B. 11

C. 18

D. 20

(3) 对动态查找有高效率的查找表组织结构是 ()。

A. 有序表

B. 分块有序表

C. 循环链表

D. B 树

(4) 一棵 3 阶 B 树中含有 2047 个关键字, 包括叶子结点层, 该树的最大深度为 ()。

A. 11

B. 12

C. 13

D. 14

(5) 已知一棵 5 阶 B 树有 53 个关键字,并且每个结点的关键字都达到最少状态,则它的深度是 (不包括叶子结点层)()。

A. 3

B. 4

C. 5

D. 6

(6) 在一棵 m 阶 B 树中, 若在某结点中插入一个新关键字而引起该结点的分裂, 则此结点中原有的关键字个数是 ()。

A. m

B. $m + 1$

C. $m - 1$

D. $m/2$

(7) 下列关于 m 阶 B 树的说法中, 说法正确的是 ()

- A. 每个结点至少有 $\lceil m/2 \rceil$ 棵非空子树
- B. 树中的结点至多有 m 个关键字
- C. 所有叶子结点都在同一层
- D. 当插入一个数据项引起 B 树结点分裂后, 树长高一层

(8) 下列关于 B 树和 $B+$ 树的叙述中,不正确的是()。

- A. B 树和 $B+$ 树都能有效地支持顺序查找
- B. B 树和 $B+$ 树都能有效地支持随机查找
- C. B 树和 $B+$ 树都是平衡多叉树
- D. B 树和 $B+$ 树都可以用于文件索引结构

(9) 下列说法中, 正确的是 ()。

- A. 在 9 阶 B 树中,除叶子以外的任意结点的分支数介于 5 和 9 之间
- B. 高度为 8 的 3 阶 B 树中关键字数最少是 255
- C. 在一棵 7 阶 B 树中,一个结点中最多有 6 棵子树,最少有 3 棵子树
- D. 平衡二叉树中,若某个结点的左、右孩子的平衡因子为零,则该结点的平衡因子一定是零

(10) 红黑树在处理过程中红黑结点会产生冲突, 请问下列解决冲突操作中, 正确的是()。

- A. 插入操作时, 解决红黑冲突
- B. 删除操作时, 解决红黑冲突
- C. 插入操作时, 解决红红冲突
- D. 删除操作时, 解决黑黑冲突

(11) 下列选项中, 正确的是 ()。

I. 红黑树是一种特殊的二叉查找树 *II*. 红黑树中的某棵子树 (若存在), 也一定是二叉查找树

III. 红黑树中的某棵子树 (若存在), 也一定是红黑树

IV. 红黑树中的某棵子树 (若存在), 一定不是红黑树

A. *I*、*II*

B. *I*、*II* 和 *III*

C. *III* 和 *IV*

D. *II*、*III* 和 *IV*

(12) 下列选项中, 错误的是()。

- A. 非空红黑树的结点要么是红色, 要么是黑色
- B. 在一棵红黑树中, 如果一个结点是黑的, 那么它的孩子结点 (如果有的话) 一定是红的
- C. 在一棵红黑树中, 如果所有结点都是黑的, 那么它的形态一定是满二叉树
- D. 红黑树的查找路径上不允许出现两个连续的红结点

(13) 下列选项中, 不正确的是()。

- A. 若红黑树黑高为 h , 则最多有 $2^{2h} - 1$ 个内部结点
- B. 若红黑树黑高为 h , 则最少有 $2^h - 1$ 个内部结点
- C. 含 n 个内部结点的红黑树, 高度不超过 $2\log_2(n+1)$
- D. 红黑树中, 红结点的数量不会超过内部结点总数的一半

(14) 下列关于红黑树的说法中, 不正确的是()。

- A. 具有 n 个关键字的红黑树中红的内部结点数与黑的内部结点数之比最大是 2:1
- B. 如果一个结点是黑色的, 则它的父结点和孩子结点都有可能是黑色的
- C. 插入 $n(n > 1)$ 个结点的形成的红黑树, 它至少有 1 个红结点
- D. 在通常情况下, 和含有相同结点数的 *AVL* 树相比, 红黑树的查询效率较优

(15) 下列关于红黑树和 *AVL* 树的描述中,不正确的是 ()。

- A. 两者都属于自平衡的二叉排序树
- B. 一个黑高是 k 的红黑树中, 内部结点最多有 $2^{2k} - 1$ 个
- C. 红黑树是一种特殊的平衡二叉树
- D. 对于红黑树而言,其任一结点的左右子树高度之差不能超过 2 倍

(16) 下列关于红黑树的说法中, 正确的是()。

- A. 红黑树插入和删除过程至多有 2 次旋转操作
- B. *AVL* 树中, 任何一个结点的左右子树高度之差不超过 1
- C. 红黑树内的红色结点的数量最多和黑色结点数目相同
- D. 红黑树的任何一棵子树也是一颗红黑树

(17) 将关键字 2,4,6,8,10,12,14 依次插入空的红黑树 T , 则 T 中有 () 个红结点。

A. 1

B. 2

C. 3

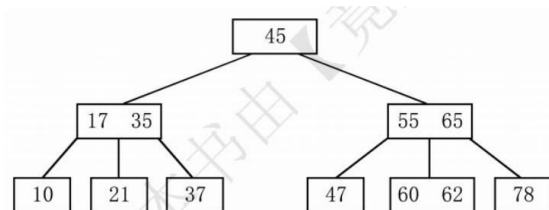
D. 4

7.4.7 B/B+树基础、红黑树-真题演练(答案见原书页码 P328)

(18) 【2009】以下叙述不符合 m 阶 B 树定义要求的是()。

- A. 根结点最多有 m 棵子树
- B. 所有叶结点都在同一层上
- C. 各结点内关键字均升序或降序排列
- D. 叶结点之间通过指针链接

(19) 【2012】已知一棵 3 阶 B 树, 如下图所示。删除关键字 78 得到一棵新 B 树, 其中最右叶结点中的关键字是 ()。



A. 60

B. 60,62

C. 62,65

D. 65

(20) 【2013】在一棵高度为 2 的 5 阶 B 树中,所含关键字的个数最少是()。

A. 5

B. 7

C. 8

D. 14

(21) 【2014】在一棵具有 15 个关键字的 4 阶 B 树中,含关键字的结点个数最多是()。

A. 5

B. 6

C. 10

D. 15

(22) 【2016】 $B+$ 树不同于 B 树的特点之一是 ()。

- A. 能支持顺序查找
- B. 结点中含有关键字
- C. 根结点至少有两个分支
- D. 所有叶结点都在同一层上

(23) 【2017】下列应用中, 适合使用 $B+$ 树的是 ()。

- A. 编译器中的词法分析
- B. 关系数据库系统中的索引
- C. 网络中的路由表快速查找
- D. 操作系统的磁盘空闲块管理

(24) 【2018】高度为 5 的 3 阶 B 树含有的关键字个数至少是 ()。

A. 15

B. 31

C. 62

D. 242

(25) 【2020】依次将关键字 5, 6, 9, 13, 8, 2, 12, 15 插入初始为空的 4 阶 B 树后,根结点中包含的关键字是()。

A. 8

B. 6,9

C. 8,13

D. 9,12

(26) 【2021】在一棵高度为 3、阶数为 3 的 B 树中,根为第 1 层,若第 2 层有 4 个关键字,则该树的结点个数最多是()。

A. 11

B. 10

C. 9

D. 8

7.5.5 散列表-习题精编(答案见原书页码 P340)

(1)*Hash* 函数有一个共同的性质,即函数值应当尽量以 () 取其值域的每个值。

- A. 最大概率 B. 最小概率 C. 平均概率 D. 同等概率

(2) 哈希表构建时采用线性探测法处理冲突,在某关键字查找成功的情况下,所探测的多个位置上的关键字 ()。

- A. 不一定都是同义词
- B. 一定是同义词
- C. 一定都不是同义词
- D. 必然有序

(3) 理论上, 散列表的平均比较次数为 ()。

A. 1

B. 2

C. 3

D. n

(4) 下面关于哈希 (*Hash*) 查找的说法正确的是 ()。

- A. 哈希函数构造越复杂越好,因为这样随机性好,冲突小
- B. 除留余数法是所有哈希函数中最好的
- C. 不存在特别好与坏的哈希函数,要视情况而定
- D. 若要在哈希表中删去一个元素,不管用什么方法解决冲突,都只需简单将该元素删去即可

(5) 下列关于散列冲突的说法中, 正确的是()。

- A. 再散列在处理冲突时不会产生“堆积”现象
- B. 在哈希表中进行查找是不需要进行关键字比较的
- C. 散列函数选的好可以减少冲突现象
- D. 采用线性探测法处理冲突时, 所有同义词在散列表中一定相邻

(6) 设哈希表长为 13, 哈希函数是 $H(key) = key \% 13$, 表中已有关键字 18, 39, 75, 93 共 4 个, 现要将关键字为 70 的结点加到表中, 用伪随机探测再散列法解决冲突, 使用的伪随机序列为 5, 8, 3, 9, 7, 1, 6, 4, 2, 11, 13, 21, 则放入的位置是 ()。

A. 8

B. 11

C. 7

D. 5

(7) 从 19 个有序元素中查找其中某元素,共进行了 6 次元素之间的比较,则采用的查找算法可能是 ()

*I.*折半查找 *II.*二叉排序树查找 *III.*顺序查找 *IV.*哈希查找

A. 仅 *I* B. *I* 和 *II* C. *II*、*III* 和 *IV* D. *II* 和 *IV*

(8) 对线性表 (7, 34, 55, 25, 64, 46, 20, 10) 进行散列存储时, 用链地址法构造散列表, 若选用 $H(key) = key \bmod 9$ 作为散列函数, 则散列地址为 1 的元素有 () 个。

A. 1

B. 2

C. 3

D. 4

(9) 设哈希表长为 12, 哈希函数为 $H(key) = key \bmod 11$, 表中已有数据关键字为 26, 16, 50, 68 共 4 个。现要将关键字为 38 的结点加到列表中, 用线性探测再散列解决冲突, 则放入的位置是 ()。

A. 3

B. 5

C. 7

D. 9

(10) 采用链地址法解决冲突的哈希表中, 查找成功的平均查找长度 ()。

- A. 直接与关键字个数有关
- B. 直接与装填因子有关
- C. 直接与表的容量有关
- D. 直接与哈希函数有关

(11) 设哈希表长 $m = 14$, 哈希函数 $H(key) = key \bmod 7$ 。表中已有 4 个结点: $ADDR(15) = 1, ADDR(38) = 3, ADDR(61) = 5; ADDR(84) = 0$, 其余地址为空。如用二次探测再用哈希法解决冲突, 关键字为 68 的结点的地址是 ()。

A. 3

B. 5

C. 6

D. 8

(12) 已知一个线性表 $(1, 13, 12, 34, 38, 33, 27, 22)$, 假定采用 $H(k) = k \% 11$ 计算散列地址进行散列存储, 若用链地址法处理冲突, 则查找成功的平均查找长度为 ()

A. 1

B. $9/8$ C. $13/11$ D. $13/8$

(13) 假定有 k 个关键字互为同义词, 若用线性探测法把这 k 个关键字存入散列表中, 至少要进行 () 次探测。

A. $k-1$ 次

B. k 次

C. $k+1$ 次

D. $k(k+1)/2$ 次

(14) 采用开放定址法解决冲突的哈希查找中, 发生集聚的原因主要是 ()。

- A. 数据元素过多
- B. 负载因子过大
- C. 哈希函数选择不当
- D. 解决冲突的算法选择不好

(15) 下列说法中正确的是 ()。

- A. 散列函数越复杂越好,因为这样随机性好,冲突概率小
- B. 在散列查找中,“比较”操作一般也是不可避免的
- C. 若散列表的负载因子小于 1, 则可避免碰撞的产生
- D. 若装填因子为 1,则向散列表中散列元素时一定会产生冲突

7.5.6 散列表-真题演练(答案见原书页码 P340)

(16) 【2011】为提高散列 (*Hash*) 表的查找效率, 可以采取的正确措施是 ()。

I. 增大装填 (载) 因子

II. 设计冲突 (碰撞) 少的散列函数

III. 处理冲突 (碰撞) 时避免产生聚集 (堆积) 现象

A. 仅 *I*

B. 仅 *II*

C. 仅 *I*、*II*

D. 仅 *II*、*III*

(17) 【2014】用哈希 (散列) 方法处理冲突 (碰撞) 时可能出现堆积 (聚集) 现象,下列选项中,会受堆积现象直接影响的是 ()。

- A. 存储效率 B. 散列函数 C. 装填 (装载) 因子 D. 平均查找长度

(18) 【2018】 现有长度为 7、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 22,43,15 依次插入到 HT 后，查找成功的平均查找长度是 ()。

A. 1.5

B. 1.6

C. 2

D. 3

(19) 【2019】 现有长度为 11 且初始为空的散列表 HT ，散列函数是 $H(key) = key \% 7$ ，采用线性探查（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) 依次插入到 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

7.6.4 字符串模式匹配-习题精编(答案见原书页码 P350)

(1) 下面关于串的叙述中,不正确的是()。

- A. 串的长度是指串中所含字符的个数
- B. 空串是由空格构成的串
- 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

(2) 若串 $S = "database"$, 其子串数目 (重复字符串算不同的) 为 ()。

A. 37

B. 36

C. 35

D. 34

(3) 设有两个串 p 和 q , 求 q 在 p 中首次出现的位置的运算称作 ()。

- A. 连接
- B. 模式匹配
- C. 求子串
- D. 求串长

(4) 下列叙述中说法错误的是()。

- A. 串的两种最基本的存储方式是顺序存储和链式存储
- B. 两个串相等的充分必要条件是两串长度相等且对应位置的字符相等
- C. 空格串与空串是相同的
- D. 串是一种特殊的线性表,其特殊性体现在数据元素是一个字符

(5) 已知字符串 S 为 "aabaacaadaafa", 模式串 P 为 "aada"。采用 KMP 算法结合 $nextval$ 数组进行匹配, 匹配成功时共进行 () 次比较。

A. 11

B. 12

C. 13

D. 14

(6) 已知串 $S = "aaab"$, 其 $next$ 数组值为 ()。

A. 0123

B. 1123

C. 1231

D. 1211

(7) 串 "ababaaababaa" 的 *next* 数组为 ()。

A. 012345678999

B. 012121111212

C. 011234223456

D. 012301232234

(8) 已知串 $S = "aabaab"$, 其 $next$ 数组值为 ()。

A. 0121233

B. 0123122

C. 0123233

D. 0121213

(9) 模式串 "abaabcac" 的 *nextval* 数组值是 ()。

A. 01122312

B. 01021302

C. 00011201

D. 01011201

(10) 已知字符串 "pqppqpp", 它的 *nextval* 数组值是 ()。

A. 01021040

B. 01021243

C. 01122240

D. 01122343

(11) 字符串 "ababaabab" 的 *nextval* 数组为 ()。

A. (0,1,0,1,0,4,1,0,1)

B. (0,1,0,1,0,2,1,0,1)

C. (0,1,0,1,0,0,0,1,1)

D. (0,1,0,1,0,1,0,1,1)

7.6.5 字符串模式匹配-真题演练(答案见原书页码 P350)

(12) 【2015】已知字符串 s 为“abaabaabacacaabaabcc”，模式串 t 为“abaabc”。采用 KMP 算法进行匹配，第一次出现“失配”($s[i] \neq t[j]$) 时， $i = j = 5$ ，则下次开始匹配时， i 和 j 的值是 ()。

A. $i = 1, j = 0$

B. $i = 5, j = 0$

C. $i = 5, j = 2$

D. $i = 6, j = 2$

(13) 【2019】 设主串 $T = "abaabaabcabaabc"$, 模式串 $S = "abaabc"$, 采用 KMP 算法进行模式匹配, 到匹配成功时为止, 在匹配过程中进行的单个字符间的比较次数是 ()。

A. 9

B. 10

C. 12

D. 15

第 8 章排序

8.2.4 插入类排序-习题精编(答案见原书页码 P364)

(1) 直接插入排序 n 个不同的元素, 最多要进行 () 次比较。

A. n

B. $n - 1$

C. $n + 1$

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

(2) 对序列 $\{12, 8, 4, 6, 21, -9, 5\}$ 进行第一趟的直接插入排序后, 得到的序列是 ()。

- A. $8, 12, 4, 6, 21, -9, 5$ B. $12, 8, 4, 6, 21, 5, -9$ C. $12, 4, 8, 6, 21, -9, 5$ D. $8, 12, 21, -9, 6, -9, 5$

(3) 下列关于直接插入排序和折半插入排序算法的说法中, 正确的有 ()。

I. 直接插入排序的平均时间复杂度为 $O(n^2)$

II. 折半插入排序的时间复杂度为 $O(n \log_2 n)$

III. 希尔排序是不稳定的排序算法

IV. 折半插入排序每趟的比较次数的数量级是确定的, 而直接插入排序每趟的比较次数不确定

A. *I, II, III, IV*

B. *I, III, IV*

C. *I, II, IV*

D. *II, III, IV*

(4) 对序列 $\{15, 9, 7, 8, 20, -1, 4\}$ 用希尔排序方法排序, 经过一趟排序后序列变为 $\{15, -1, 4, 8, 20, 9, 7\}$, 则该次采用的增量是()。

A. 1

B. 4

C. 3

D. 2

(5) 用希尔排序对数组 $\{98, 36, -9, 0, 47, 23, 1, 8, 10, 7\}$ 进行排序, 给出的步长 (也称增量序列) 依次是 4, 2, 1, 第一趟结束后数组中数据的排序次序 ()。

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. 10, 7, -9, 47, 0, 23, 1, 8, 98, 36

8.2.5 插入类排序-真题演练(答案见原书页码 P364)

(6) 【2012】同一待排序序列分别进行折半插入排序和直接插入排序, 两者之间可能不同之处是 ()

- A. 排序总趟数
- B. 元素的移动次数
- C. 使用辅助空间的数量
- D. 元素之间的比较次数

(7) 【2014】用希尔排序方法对一个数据序列进行排序时, 若第 1 趟排序结果为 {9, 1, 4, 13, 7, 8, 20, 23, 15}, 则该趟排序采用的增量 (间隔) 可能是 ()。

A. 2

B. 3

C. 4

D. 5

(8) 【2015】希尔排序的组内排序采用的是()。

- A. 直接插入排序 B. 折半插入排序 C. 快速排序 D. 归并排序

(9) 【2018】对初始数据序列 (8,3,9,11,2,1,4,7,5,10,6) 进行希尔排序。若第一趟排序结果为 (1,3,7,5,2,6,4,9,11,10,8), 第二趟排序结果为 (1,2,6,4,3,7,5,8, 11,10,9), 则两趟排序采用的增量 (间隔) 依次是 ()。

A. 3,1

B. 3,2

C. 5,2

D. 5,3

8.3.3 交换类排序-习题精编(答案见原书页码 P372)

(1) 对数据序列 $\{8, 9, 10, 4, 5, 6, 20, 1, 2\}$ 采用冒泡排序 (从后向前次序进行, 要求升序), 需要进行的趟数至少是 ()。

A. 3

B. 4

C. 5

D. 6

(2) 在一次遍历比较序列中找最大值、最小值, 最大值放在最右端, 最小的放在最左端, 同样缩小范围再次比较, 放在次右端、次左端, 对数组 4,7,8,3,5,6,10,9,1,2 进行双向起泡排序, 求排序趟数 ()。

A. 7

B. 6

C. 8

D. 9

(3) 一组记录的关键码为 (46, 79, 56, 38, 40, 84), 则利用快速排序的方法, 以第一个记录为基准, 从小到大得到的一次划分结果为 ()。

- A. (38, 40, 46, 56, 79, 84)
- B. (40, 38, 46, 79, 56, 84)
- C. (40, 38, 46, 56, 79, 84)
- D. (40, 38, 46, 84, 56, 79)

(4) 下列关于快速排序的说法中, 正确的是()。

*I.*快速排序算法的平均时间复杂度为 $O(n\log_2 n)$

*II.*快速排序是一个不稳定的排序算法

*III.*快速排序算法的性能关键在于划分操作的好坏

*IV.*快速排序的空间复杂度平均为 $O(1)$

A. *I, II, III*

B. *I, II, IV*

C. *II, III*

D. *I, III, IV*

(5) n 个关键字进行快速排序, 最大递归深度为 (), 最小递归深度为 ()。

A. 1

B. n

C. $\log_2 n$

D. $n \log_2 n$

(6) 下列序列中, 可能是执行第一趟快速排序后所得到的序列 ()。

A. 68, 11, 18, 69, 23, 93, 73

B. 93, 73, 68, 11, 69, 23, 18

C. 68, 73, 93, 11, 69, 23, 18

D. 68, 11, 69, 23, 18, 93, 73

(7) 快速排序在最坏情况下的时间复杂度与下列哪个算法最坏情况下的时间复杂度相同()。

A. 希尔排序

B. 堆排序

C. 冒泡排序

D. 基数排序

(8) 对下列关键字序列用快速排序法进行排序时,速度最快的情形是()。

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

8.3.4 交换类排序-真题演练(答案见原书页码 P372)

(10) 【2010】用递归方式对顺序表进行快速排序。下列关于递归次数的叙述中, 正确的是()。

- A. 递归次数与初始数据的排列次序无关
- B. 每次划分后, 先处理较长的分区可以减少递归次数
- C. 每次划分后, 先处理较短的分区可以减少递归次数
- D. 递归次数与每次划分后得到的分区的处理顺序无关

(11) 【2011】为实现快速排序算法,待排序序列宜采用的存储方式是()。

- A. 顺序存储 B. 散列存储 C. 链式存储 D. 索引存储

(12) 【2014】下列选项中, 不可能是快速排序第 2 趟排序结果的是 ()。

- A. 2,3,5,4,6,7,9 B. 2,7,5,6,4,3,9 C. 3,2,5,4,7,6,9 D. 4,2,3,5,7,6,9

(13) 【2019】排序过程中, 对尚未确定最终位置的所有元素进行一遍处理称为一“趟”。下列序列中, 不可能是快速排序第二趟结果的是()。

A. 5, 2, 16, 12, 28, 60, 32, 72

B. 2, 16, 5, 28, 12, 60, 32, 72

C. 2, 12, 16, 5, 28, 32, 72, 60

D. 5, 2, 12, 28, 16, 32, 72, 60

8.4.3 选择类排序-习题精编(答案见原书页码 P384)

(1) 在以下排序算法中, 每次从未排序的记录中选取最小关键字的记录, 加入已排序记录的末尾, 该排序方法是 ()。

A. 简单选择排序

B. 冒泡排序

C. 堆排序

D. 直接插入排序

(2) 对 n 个记录的文件进行堆排序, 最坏情况下的执行时间是 ()。

A. $O(\log_2 n)$

B. $O(n)$

C. $O(n \log_2 n)$

D. $O(n^2)$

(3) 一组关键字为 (46, 79, 56, 38, 40, 84), 则利用堆排序的方法建立大顶堆的初始堆为 ()。

A. 79, 46, 56, 38, 40, 84

B. 84, 79, 56, 38, 40, 46

C. 84, 79, 56, 46, 40, 38

D. 84, 79, 56, 46, 38, 40

(4) 有一组数据 $(15, 9, 7, 8, 20, -1, 7, 4)$, 用堆排序的筛选方法建立的初始小根堆为 ()。

A. $-1, 4, 8, 9, 20, 7, 15, 7$

B. $-1, 7, 15, 7, 4, 8, 20, 9$

C. $-1, 4, 7, 8, 20, 15, 7, 9$

D. $A、B、C$ 都不符合

(5) 下列哪一个序列可以是一个堆 ()。

A. 20,76,35,27,98,57

B. 98,27,35,76,20,57

C. 20,57,27,98,35,76

D. 20,35,27,98,57,76

(6) 对序列 (22,16,71,59,24,7,67,70,51) 进行堆排序形成小根堆, 删除 2 个堆顶元素后的剩余小根堆是 ()。

A. 22,71,59,24,67,70,51

B. 22,24,51,59,70,71,67

C. 70,24,22,51,59,71,67

D. 22,24,67,51,59,71,70

(7) 已知大根堆 (62, 34, 53, 12, 8, 46, 22) 删除堆顶元素后需重新调整堆, 此过程中关键字的比较次数为()。

A. 2

B. 3

C. 4

D. 5

(8) 哪种数据结构是从根结点到任意叶结点的路径都是有序的?()。

A. AVL

B. 二叉查找树

C. 哈夫曼树

D. 堆

(9) 对相同的 n 个整数构成的二叉排序树和小根堆, 下面那个说法是不正确的?()。

- A. 二叉排序树高度大于等于小根堆高度
- B. 对该二叉排序树进行中序遍历可得到从小到大的序列
- C. 从小根堆根结点到其任何叶结点的路径上的结点值构成从小到大的序列
- D. 对该小根堆进行按层次 (*levelorder*) 遍历可得到从小到大的序列

8.4.4 选择类排序-真题演练(答案见原书页码 P384)

(10) 【2009】已知关键字序列 (5, 8, 12, 19, 28, 20, 15, 22) 是小根堆 (最小堆), 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

(11) 【2011】已知序列 (25,13,10,12,9) 是大根堆, 在序列尾部插入新元素 18, 将其再调整为大根堆, 调整过程中元素之间进行的比较次数是 ()。

A. 1

B. 2

C. 4

D. 5

(12) 【2015】 已知小根堆为 (8, 15, 10, 21, 34, 16, 12), 删除关键字 8 之后需重建堆, 在此过程中, 关键字之间的比较次数是 ()。

A. 1

B. 2

C. 3

D. 4

(13) 【2018】在将数据序列 (6,1,5,9,8,4,7) 建成大根堆时, 正确的序列变化过程是 ()。

- A. 6,1,7,9,8,4,5 \rightarrow 6,9,7,1,8,4,5 \rightarrow 9,6,7,1,8,4,5 \rightarrow 9,8,7,1,6,4,5
- B. 6,9,5,1,8,4,7 \rightarrow 6,9,7,1,8,4,5 \rightarrow 9,6,7,1,8,4,5 \rightarrow 9,8,7,1,6,4,5
- C. 6,9,5,1,8,4,7 \rightarrow 9,6,5,1,8,4,7 \rightarrow 9,6,7,1,8,4,5 \rightarrow 9,8,7,1,6,4,5
- D. 6,1,7,9,8,4,5 \rightarrow 7,1,6,9,8,4,5 \rightarrow 7,9,6,1,8,4,5 \rightarrow 9,7,6,1,8,4,5 \rightarrow 9,8,6,1,7,4,5

(14) 【2020】 以下关于大根堆 (至少含 2 个元素) 的叙述中正确的是 ()。

I. 可以将堆看成一棵完全二叉树 *II.* 可以采用顺序存储方式保存堆

III. 可以将堆看成一棵二叉排序树 *IV.* 堆中的次大值一定在根的下一层

A. *I, II*

B. *II, III*

C. *I, II, IV*

D. *I, III, IV*

(15) 【2021】将关键字 6,9,1,5,8,4,7 依次插入到初始为空的大根堆 H , 得到的 H 是()。

A. 9,8,7,6,5,4,1

B. 9,8,7,5,6,1,4

C. 9,8,7,5,6,4,1

D. 9,6,7,5,8,4,1

8.5.3 二路归并排序和基数排序-习题精编(答案见原书页码 P394)

(1) 若只进行 3 趟排序就完成对 64 个元素的多路归并排序, 则选取的归并路数最少是 ()。

A. 2

B. 3

C. 4

D. 5

(2) 一组经过第一趟 2 路归并排序后的记录的关键字为 {25,50,15,35,80,85,20,40,36,70}, 其中包含 5 个长度为 2 的有序表, 用 2 路归并排序方法对该序列进行第二趟归并后的结果为()。

- A. 15,25,35,50,80,20,85,40,70,36
- B. 15,25,35,50,20,40,80,85,36,70
- C. 15,25,50,35,80,85,20,36,40,70
- D. 15,25,35,50,80,20,36,40,70,85

(3) 对 2 个各 N 个元素的有序子表进行合并, 最多的比较次数是 ()。

A. N

B. $N-1$

C. $2N$

D. $2N-1$

(4) 对关键字序列 {05,46,13,55,94,17,42} 进行基数排序, 第一趟排序的结果是 ()。

A. 05,46,13,55,94,17,42

B. 05,13,17,42,46,55,94

C. 42,13,94,05,55,46,17

D. 05,13,46,55,17,42,94

8.5.4 二路归并排序和基数排序-真题演练(答案见原书页码 P394)

(6) 【2013】对给定的关键字序列 {110, 119, 007, 911, 114, 120, 122} 进行基数排序, 则第 2 趟分配收集后得到的关键字序列是 ()。

A. 007, 110, 119, 114, 911, 120, 122

B. 007, 110, 119, 114, 911, 122, 120

C. 007, 110, 911, 114, 119, 120, 122

D. 110, 120, 911, 122, 114, 007, 119

(7) 【2017】在内部排序中, 若选择了归并排序而没有选择插入排序, 则可能的理由是()。

*I.*归并排序的程序代码更短

*II.*归并排序的占用空间更少

*III.*归并排序的运行效率更高

A. 仅 *II*

B. 仅 *III*

C. 仅 *I*、*II*

D. 仅 *I*、*III*

(8) 【2021】 设数组 $S = (93, 946, 372, 9, 146, 151, 301, 485, 236, 327, 43, 892)$, 采用最低位优先 (*LSD*) 基数排序将 S 排列成升序序列, 第一趟分配收集后, 在元素 372 之前, 之后相邻的元素是 ()。

A. 43, 892

B. 236, 301

C. 301, 892

D. 485, 301

8.6.3 各种内部排序算法总结-习题精编(答案见原书页码 P399)

(1) 对于同等大小的不同的初始序列,总的比较次数一定的是()。

- | | |
|----------------|--------------|
| A. 折半插入排序和选择排序 | B. 基数排序和归并排序 |
| C. 冒泡排序和快速排序 | D. 堆排序 |

(2) 对于不同的初始序列,总的排序趟数会改变的算法有()。

A. 冒泡排序和快速排序

B. 基数排序和归并排序

C. 插入排序

D. 堆排序

(3) 对各种内部排序方法来说 ()。

- A. 快速排序时间性能最佳
- B. 基数排序和归并排序是稳定的排序方法
- C. 快速排序是一种选择排序
- D. 堆排序所用的辅助空间较大

(4) 若需要在 $O(n\log_2 n)$ 的时间内完成对一个数组的排序, 且要求排序是稳定的, 则可选的排序方法是 ()。

A. 快速排序

B. 归并排序

C. 堆排序

D. 直接插入排序

(5) 在下列的排序算法中, 所用辅助空间最大的是()。

A. 归并排序

B. 快速排序

C. 堆排序

D. 希尔排序

(6) 在下列排序算法中, 平均时间复杂度为 $O(n\log_2 n)$ 的是 ()。

- A. 直接插入排序 B. 快速排序 C. 冒泡排序 D. 基数排序

(7) 在下列排序算法中, 需要线性辅助空间为 $O(n)$ 的是 ()。

A. 堆排序

B. 快速排序

C. 归并排序

D. 希尔排序

(8) 若只想得到 1000 个元素组成的序列中第 5 个最小元素之前的部分排序的序列,能最快得到结果的排序算法是()。

A. 冒泡排序

B. 快速排序

C. 希尔排序

D. 堆排序

(9) 以下排序方法中, 在一趟结束后不一定能选出一个元素放在其最终位置上的的是()。

- A. 简单选择排序 B. 冒泡排序 C. 归并排序 D. 堆排序

(10) 若要求排序是稳定的,且关键字为实数,则在下列排序方法中应选()。

- A. 直接插入排序 B. 简单选择排序 C. 基数排序 D. 快速排序

(11) 以下排序方法中时间复杂度为 $O(n\log_2 n)$ 且稳定的是 ()。

A. 堆排序

B. 快速排序

C. 归并排序

D. 直接插入排序

(12) 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的元素比较次数最少, 则应该采用的方法是()。

- A. 直接插入排序 B. 简单选择排序 C. 希尔排序 D. 冒泡排序

(13) 对一个初始状态为递增的序列进行按递增顺序的排序,用()最省时,用()最费时。

- A. 直接插入排序 B. 堆排序 C. 快速排序 D. 归并排序

(14) 下列排序算法中, 哪一种可能出现这种情况: 在最后一趟开始前, 所有关键字都不在其最终位置 ()。

A. 堆排序

B. 冒泡排序

C. 快速排序

D. 折半插入排序

(15) 对一组关键字 (84,47,25,15,21) 排序, 关键字的排列次序在排序过程中的变化为:(84, 47,25,15, 21),(15,47,25,84,21),(15,21,25,84,47),(15,21,25,47,84),则采用的排序是()。

- A. 简单选择排序 B. 冒泡排序 C. 快速排序 D. 直接插入排序

(16) 关键字序列 (8,9,10,4,5,6,20,1,2) 只能是哪一种算法中两趟排序后的结果 ()。

- A. 简单选择排序 B. 冒泡排序 C. 直接插入排序 D. 堆排序

(17) 对于关键字序列 $(15, 9, 7, 8, 20, -1, 4)$ 进行排序, 进行一趟后关键字序列变为 $(9, 15, 7, 8, 20, -1, 4)$, 则采用的哪一种算法 ()。

A. 简单选择排序

B. 冒泡排序

C. 堆排序

D. 直接插入排序

(18) 如果一台计算机具有多核 *CPU*, 可以同时执行相互独立的任务。归并排序的各个归并段也可以并行执行, 因此称归并排序是可以并行执行的。以下排序方法不可以并行执行的有 ()。

I. 基数排序 *II*. 快速排序 *III*. 冒泡排序 *IV*. 堆排序

A. *I*、*III* B. *I*、*II* C. *I*、*III*、*IV* D. *II*、*IV*

(19) 设线性表中每个元素有两个数据项 k_1 和 k_2 , 现对线性表按下列规则进行排序: 先看数据项 k_1 , k_1 值小的在前, 大的在后; 在 k_1 值相同的情况下, 再看数据项 k_2 , k_2 值小的在前, 大的在后。满足这种要求的排序方法是 ()。

- A. 先按 k_1 值进行直接插入排序, 再按 k_2 值进行简单选择排序。
- B. 先按 k_2 值进行直接插入排序, 再按 k_1 值进行简单选择排序。
- C. 先按 k_1 值进行简单选择排序, 再按 k_2 值进行直接插入排序。
- D. 先按 k_2 值进行简单选择排序, 再按 k_1 值进行直接插入排序。

8.6.4 各种内部排序算法总结-真题演练(答案见原书页码 P399)

(20) 【2009】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. 插入排序

C. 选择排序

D. 二路归并排序

(21) 【2010】对一组数据 (2,12,16,88,5,10) 进行排序, 若前三趟排序结果如下。第一趟排序结果:2, 12,16,5,10,88; 第二趟排序结果:2,12,5,10,16,88; 第三趟排序结果:2, 5,10,12,16,88。则采用的方法可能是 ()。

- A. 冒泡排序 B. 希尔排序 C. 归并排序 D. 基数排序

(22) 【2012】在排序过程中, 对尚未最终确认最终位置的所有元素进行一遍处理称为一趟排序。下列排序方法中, 每一趟排序结束都至少能够确定一个元素的最终位置的方法()。

*I.*简单选择排序 *II.*希尔排序 *III.*快速排序 *IV.*堆排序 *V.*二路归并排序

A. 仅 *I*、*III*、*IV* B. 仅 *I*、*III*、*V* C. 仅 *II*、*III*、*IV* D. 仅 *III*、*IV*、*V*

(23) 【2015】下列排序算法中, 元素的移动次数与关键字的初始排列次序无关的是()。

- A. 直接插入排序 B. 冒泡排序 C. 基数排序 D. 快速排序

(24) 【2017】下列排序方法中, 若将顺序存储更换为链式存储, 则算法的时间效果会降低的是()。

*I.*插入排序 *II.*选择排序 *III.*冒泡排序 *IV.*希尔排序 *V.*堆排序

A. 仅 *I*、*II* B. 仅 *II*、*III* C. 仅 *III*、*IV* D. 仅 *IV*、*V*

(25) 【2019】选择一个排序算法时,除算法的时空效率外,下列因素中,还需要考虑的是()。

*I.*数据的规模 *II.*数据的存储方式 *III.*算法的稳定性 *IV.*数据的初始状态

A. 仅 *III* B. 仅 *I*、*II* C. 仅 *II*、*III*、*IV* D. *I*、*II*、*III*、*IV*

(26) 【2020】对大部分元素已有序的数组进行排序时, 直接插入排序比简单选择排序效率更高, 其原因是 ()。

*I.*直接插入排序过程中元素之间的比较次数更少

*II.*直接插入排序过程中所需要的辅助空间更少

*III.*直接插入排序过程中元素的移动次数更少

A. *I*

B. *III*

C. *I, II*

D. *I, II, III*

8.7.5 外部排序-习题精编(答案见原书页码 P410)

(1) 下列关于小顶堆和败者树的说法正确的是()。

I. 败者树是从下往上维护, 每上一层, 只需和败者节点比较 1 次即可

II. 堆在维护时从上往下, 每下一层, 需和左右子节点都比较, 需要比较 2 次

III. 败者树内的结点存储的内容是对应元素数值

A. I, III

B. II, III

C. I, II, III

D. I, II

(2) 下列关于小顶堆和败者树的说法正确的是 ()。

I. 败者树每一次维护, 必定要从叶子结点一直走到根结点, 不可能从中间停止

II. 堆维护一个结点位置的过程中必定要从根结点一直走到叶结点, 不可能从中间停止

III. 对于相同规模的处理对象, 败者树的构建需要的空间是小顶堆的 2 倍

A. *I, III*

B. *II, III*

C. *I, II, III*

D. *I, II*

(3) 下列关于选择－置换排序的说法正确的是()。

*I.*创建初始段文件过程中, 需要不断地在内存工作区中选择不小于旧的 *MINIMAX* 的最小值, 此过程需利用败者树实现

*II.*不断选择新的 *MINIMAX* 记录时, 为防止新加入的关键字值更小, 每个叶结点都附加一个序号位, 当进行关键字比较时, 先比较序号, 序号大的为胜者; 序号相同的关键字值小的为胜者

*III.*置换选择排序算法得到的初始归并段的长度可以超过内存容量限制, 且获得的归并段的平均长度为内存工作区大小的两倍

A. *I, III*

B. *II, III*

C. *I, II, III*

D. *I, II*

(4) 设内存工作区能容纳 m 个记录, 那么对磁盘上的 n 个记录进行 k 路平衡归并排序, 需要做多少趟归并排序 ()。

A. $\log_k n/m$

B. kmn

C. $\log_k mn$

D. nm/k

(5) 若初始归并段大小分别为:5,9,12,13,14,16,17,18,20,28,30,37,42, 采用四路归并, 那么最佳归并树的带权路径长度 WPL 是()。

A. 460

B. 472

C. 480

D. 486

(6) 下列说法中正确的是()。

I. 置换选择排序的作用是生成初始归并段

II. 最佳归并树的作用是设计多路归并排序的优化方案

III. 外部排序是指排序前后记录在外存,排序时记录调入内存的排序算法

A. *I, II, III*

B. *I*

C. *I, II*

D. *II, III*

(7) 下面关于外部排序的说法中正确的是()。

- A. 外部排序之前应该生成归并段
- B. 外部排序不需要使用内存
- C. 置换选择排序的作用是完成归并排序
- D. 最佳归并树用来生成初始归并段

(8) 下列关于外部排序的说法中正确的是()。

- A. 置换选择排序得到的初始归并段的长度一定相等
- B. 内存与外存交换信息的时间只占一小部分
- C. 败者树是一颗完全二叉树
- D. 外部排序适合小数据量

8.7.6 外部排序-真题演练(答案见原书页码 P410)

(9) 【2016】对 $10TB$ 的数据文件进行排序, 应使用的方法是 ()。

- A. 希尔排序 B. 堆排序 C. 快速排序 D. 归并排序

(10) 【2019】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, 需要补充的虚段个数是()。

A. 1

B. 2

C. 3

D. 4