

# 目录

第 1 章绪论.....	3
第 2 章线性表.....	6
2.1 线性表的顺序存储.....	6
2.2 线性表的链式存储.....	11
第 3 章栈、队列和数组.....	17
3.1 栈.....	17
3.2 队列.....	19
3.3 栈与队列的应用.....	22
3.4 数组与压缩存储.....	23
第 4 章树与二叉树.....	25
4.1 树的基本概念.....	25
4.2 二叉树基础.....	25
4.3 二叉树的遍历.....	27
4.4 线索二叉树.....	31
4.5 树与森林.....	32
4.6 树与二叉树的应用.....	33
第 5 章图.....	39
5.1 图的基本概念.....	39
5.2 图的基本存储及基本操作.....	40
5.3 图的遍历.....	43
5.4 图的基本应用.....	44
第 6 章查找.....	56
6.1 顺序查找、折半查找与分块查找.....	56
6.2 二叉搜索树、平衡二叉树和红黑树.....	58
6.3 B、B+树基础.....	63
6.4 散列(Hash)表.....	65

6.5 字符串匹配模式 .....	68
第 7 章排序 .....	69
7.1 插入类排序 .....	69
7.2 交换类排序 .....	70
7.3 选择类排序 .....	72
7.4 二路归并排序和基数排序 .....	75
7.5 各种内部排序算法总结 .....	76
7.6 外部排序 .....	80

## 第 1 章绪论

1. 【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( )。 【P20, 01 题】

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

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

2. 【2012】 求整数 $n(n \geq 0)$ 阶乘的算法如下, 其时间复杂度是( )。 【P29, 01 题】

```
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.  $O(n^2)$

3. 【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( )。 【P39, 01 题】

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

4. 【2014】下列程序段的时间复杂度是( )。

【P48, 01 题】

```
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.  $O(n^2)$

5. 【2017】下列函数的时间复杂度是( )。

【P76, 01 题】

```
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.  $O(n \log_2 n)$

6. 【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( )。

【P95, 01 题】

```
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.  $O(n^2)$

7. 【2022】下列程序段的时间复杂度是( )。

【P124, 01 题】

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

A.  $O(\log_2 n)$

B.  $O(n)$

C.  $O(n\log_2 n)$

D.  $O(n^2)$

## 第 2 章线性表

### 2.1 线性表的顺序存储

1. 【2010】 设将 $n(n > 1)$ 个整数存放到一维数组 $R$ 中。试设计一个在时间和空间两方面都尽可能高效的算法。将 $R$ 中保存的序列循环左移 $p(0 < p < n)$ 个位置,即将 $R$ 中的数据由 $(X_0, X_1, \dots, X_{n-1})$ 变换为 $(X_p, X_{p+1}, \dots, X_{n-1}, X_0, X_1, \dots, X_{p-1})$ 。要求: 【P16, 42 题】

- (1) 给出算法的基本设计思想;
- (2) 根据设计思想,采用C或C++或Java语言描述算法,关键之处给出注释;
- (3) 说明你所设计算法的时间复杂度和空间复杂度。

2. 【2011】一个长度为 $L(L \geq 1)$ 的升序序列 $S$ , 处在第 $\lfloor L/2 \rfloor$ 个位置的数称为 $S$ 中位数。例如, 若序列 $S_1 = (11, 13, 15, 17, 19)$ , 则 $S_1$ 的中位数是 15。两个序列的中位数是含它们所有元素的升序序列的中位数。例如, 若 $S_2 = (2, 4, 6, 8, 20)$ , 则 $S_1$ 和 $S_2$ 的中位数是 11。现有两个等长的升序序列 $A$ 和 $B$ , 试设计一个在时间和空间两方面都尽可能高效的算法, 找出两个序列 $A$ 和 $B$ 的中位数。

要求:

【P25, 42 题】

- (1) 给出算法的基本设计思想;
- (2) 根据设计思想, 采用C或C++或Java语言描述算法, 关键处给出注释;
- (3) 说明你所设计算法的时间复杂度和空间复杂度。

3. 【2013】已知一个整数序列  $A = (a_0, a_1, a_2, \dots, a_n)$ , 其中  $0 \leq a_i < n (0 \leq i < n)$ 。若存在  $a_{p_1} = a_{p_2} = \dots = a_{p_m} = x$  且  $m > n/2 (0 \leq p_k < n, 1 \leq k \leq m)$ , 则称  $x$  为  $A$  的主元素。例如  $A = (0, 5, 5, 3, 5, 7, 5, 5)$ , 则 5 为主元素; 又如  $A = (0, 5, 5, 3, 5, 1, 5, 7)$ , 则  $A$  中没有主元素。假设  $A$  中的  $n$  个元素保存在一个一维数组中, 请设计一个尽可能高效的算法, 找出  $A$  的主元素。若存在主元素, 则输出该元素; 否则输出 -1。

要求:

【P44, 41 题】

- (1) 给出算法的基本设计思想;
- (2) 根据设计思想, 采用 C 或 C++ 或 Java 语言描述算法, 关键之处给出注释;
- (3) 说明你所设计算法的时间复杂度和空间复杂度。



4. 【2018】给定一个含 $n(n \geq 1)$ 个整数的数组, 请设计一个在时间上尽可能高效的算法, 找出数组中未出现的最小正整数。例如, 数组 $\{-5, 3, 2, 3\}$ 中未出现的最小正整数是 1; 数组 $\{1, 2, 3\}$ 中未出现的最小正整数是 4。要求: 【P89, 41 题】

- (1) 给出算法的基本设计思想;
- (2) 根据设计思想, 采用C或C++语言描述算法, 关键之处给出注释;
- (3) 说明你所设计算法的时间复杂度和空间复杂度。

5. 【2020】定义三元组 $(a, b, c)$  ( $a, b, c$ 均为整数) 的距离 $D = |a - b| + |b - c| + |c - a|$ 。给定 3 个非空整数集合 $S1, S2, S3$ , 按升序分别存储在 3 个数组中。请设计一个尽可能高效的算法, 计算并输出所有可能的三元组 $(a, b, c)$  ( $a \in S1, b \in S2, c \in S3$ ) 中的最小距离。例如 $S1 = (-1, 0, 9)$ ,  $S2 = (-25, -10, 10, 11)$ ,  $S3 = (2, 9, 17, 30, 41)$ , 则最小距离为 2, 相应的三元组为 $(9, 10, 9)$ 。要求:

(1) 给出算法的基本设计思想;

【P110, 41 题】

(2) 根据设计思想, 采用C或C++语言描述算法, 关键之处给出注释;

## 2.2 线性表的链式存储

1. 【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( )。 【P39, 01 题】

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

2. 【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( )。 【P67, 01 题】

- A. 1010H, 1014H, 1004H                      B. 1010H, 1004H, 1014H  
C. 1014H, 1010H, 1004H                      D. 1014H, 1004H, 1010H

3. 【2016】已知一个带有表头结点的双向循环链表 $L$ ,结点结构为 $|\text{prev}| \text{data} | \text{next}|$ ,其中,  $\text{prev}$  和  $\text{next}$  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( )。 【P67, 02 题】

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

4. 【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: $[\text{data}|\text{next}]$ , 其中 $|\text{next}|$ 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( )。 【P115, 01 题】

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

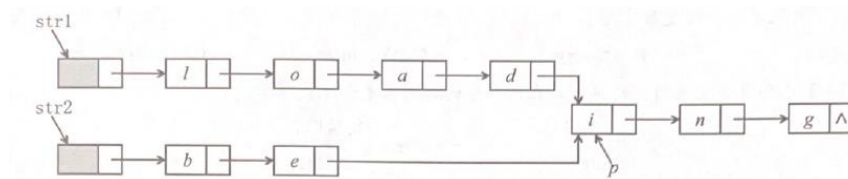
5. 【2009】已知一个带有表头结点的单链表, 结点结构为:  $1data|1link|$ 。假设该链表只给出了头指针  $list$ 。在不改变链表的前提下, 请设计一个尽可能高效的算法, 查找链表中倒数第  $k$  个位置上的结点 ( $k$  为正整数)。若查找成功, 算法输出该结点的  $data$  域的值, 并返回 1; 否则, 只返回 0。

要求:

【P6, 42 题】

- (1) 描述算法的基本设计思想。
- (2) 描述算法的详细实现步骤。
- (3) 根据设计思想和实现步骤, 采用程序设计语言描述算法 (使用 C 或 C++ 或 Java 语言实现), 关键之处请给出简要注释。

6. 【2012】假定采用带头结点的单链表保存单词,当两个单词有相同的后缀时,则可共享相同的后缀存储空间,例如,“loading”和“being”的存储映像如下图所示。

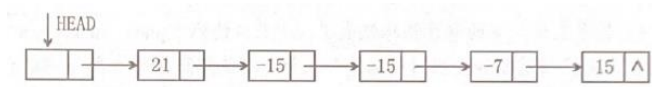


设str1和str2分别指向两个单词所在单链表的头结点,链表结点结构为:`[data|next]`。请设计一个时间上尽可能高效的算法,找出由 str1 和 str2 所指向两个链表共同后缀的起始位置(如图中字符*i*所在结点的位置*p*)。要求: 【P35, 42 题】

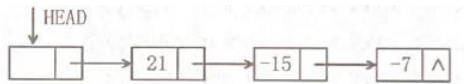
- (1) 给出算法的基本设计思想;
- (2) 根据设计思想,采用C或C++语言描述算法,关键之处给出注释;
- (3) 说明你所设计算法的时间复杂度。

7. 【2015】用单链表保存 $m$ 个整数, 结点的结构为:  $1data|link|$ , 且 $|data| \leq n$  ( $n$ 为正整数)。现要求设计一个时间复杂度尽可能高效的算法, 对于链表中  $data$  的绝对值相等的结点, 仅保留第一次出现的结点而删除其余绝对值相等的结点。

例如, 若给定的单链表 HEAD 如下:



则删除结点后的 HEAD 为:



要求:

【P63, 41 题】

- (1) 给出算法的基本设计思想。
- (2) 使用C或C++语言, 给出单链表结点的数据类型定义;
- (3) 根据设计思想, 采用C或C++语言描述算法, 关键之处给出注释。
- (4) 说明你所设计算法的时间复杂度和空间复杂度。

8. 【2019】设线性表 $L = (a_1, a_2, a_3, \dots, a_{n-2}, a_{n-1}, a_n)$ 采用带头结点的单链表保存,链表中结点定义如下:

```
typedef struct node{
    int data;
    struct node *next;
}NODE;
```

请设计一个空间复杂度为 $O(1)$ 且时间上尽可能高效的算法,重新排列 $L$ 中的各结点,得到线性表 $L' =$

$(a_1, a_n, a_2, a_{n-1}, a_3, a_{n-2}, \dots)$ 。要求:

【P101, 41 题】

- (1) 给出算法的基本设计思想;
- (2) 根据设计思想,采用C或C++语言描述算法,关键之处给出注释;
- (3) 说明你所设计的算法的时间复杂度。



## 第 3 章栈、队列和数组

### 3.1 栈

1. 【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 ( )。 【P1, 02 题】

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

2. 【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( )。 【P11, 01 题】

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

3. 【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( )。 【P20, 02 题】

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

4. 【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( )。 【P39, 02 题】

- A.  $n - 3$                       B.  $n - 2$                       C.  $n - 1$                       D. 无法确定

5. 【2017】下列关于栈的叙述中, 错误的是( )。 【P76, 02 题】

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. 只要确定了入栈次序, 就可确定出栈次序

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

6. 【2018】若栈 $S_1$ 中保存整数, 栈 $S_2$ 中保存运算符, 函数 $F(\text{栈})$ 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) 执行相应的运算 $bop a$ ;

(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 $F(\text{栈})$ 后,  $S_1$ 栈顶保存的值是( )。 【P85, 01 题】

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

7. 【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( )。 【P105, 02 题】
- A.  $b, a, c$                       B.  $b, a, e$                       C.  $b, c, a$                       D.  $b, c, e$

8. 【2022】给定有限符号集S, in 和 out 均为S中所有元素的任意排列。对于初始为空的栈ST, 下列叙述中, 正确的是( )。 【P124, 02 题】
- A. 若 in 是 ST 的入栈序列, 则不能判断 out 是否为其可能的出栈序列
- B. 若 out 是 ST 的出栈序列, 则不能判断 in 是否为其可能的入栈序列
- C. 若 in 是 ST 的入栈序列, out 是对应 in 的出栈序列, 则 in 与 out 一定不同
- D. 若 in 是 ST 的入栈序列, out 是对应 in 的出栈序列, 则 in 与 out 可能互为倒序

## 3.2 队列

1. 【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( )。 【P11, 02 题】
- A. bacde                      B. dbace                      C. dbcae                      D. ecbad

2. 【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( )。 【P20, 03 题】

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

3. 【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。初始时空。下列判断队空和队满的条件中, 正确的是( )。 【P48, 03 题】

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

4. 【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( )。 【P85, 02 题】

- 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



### 3.3 栈与队列的应用

1. 【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( )。 【P1, 01 题】

- A. 栈                      B. 队列                      C. 树                      D. 图

2. 【2012】已知操作符包括已知操作符包括 ‘+’ 、 ‘-’ 、 ‘\*’ 、 ‘/’ 、 ‘(’ 和 ‘)’ 。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( )。 【P29, 02 题】

- A. 5                      B. 7                      C. 8                      D. 11

3. 【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( )。 【P48, 02 题】

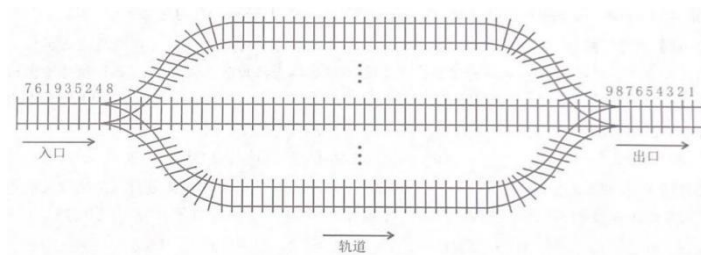
- A. + (\* -                      B. + (- \*                      C. / + (\* - \*                      D. / + - \*

4. 【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( )。 【P58, 01 题】

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

- A.  $\text{main}(\boxed{\phantom{0}}) \rightarrow S(1) \rightarrow S(0)$                       B.  $S(0) \rightarrow S(1) \rightarrow \text{main}(\boxed{\phantom{0}})$   
C.  $\text{main}(\boxed{\phantom{0}}) \rightarrow S(0) \rightarrow S(1)$                       D.  $S(1) \rightarrow S(0) \rightarrow \text{main}(\boxed{\phantom{0}})$

5. 【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( )。 【P67, 03 题】



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

### 3.4 数组与压缩存储

1. 【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( )。 【P67, 04 题】

- A. 86                      B. 87                      C. 88                      D. 89

2. 【2017】适用于压缩存储稀疏矩阵的两种存储结构是( )。 【P76, 03 题】

- 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

3. 【2018】设有一个 $12 \times 12$ 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( )。 【P85, 03 题】

- A. 50  
B. 51  
C. 55  
D. 66

4. 【2020】将一个 $10 \times 10$ 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( )。 【P105, 01 题】

- A. 15  
B. 16  
C. 22  
D. 23

5. 【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( )。 【P115, 03 题】

- A. 295  
B. 300  
C. 301  
D. 306



## 第 4 章树与二叉树

### 4.1 树的基本概念

1. 【2010】在一棵度为 4 的树  $T$  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( )。 【P11, 05 题】

- A. 41                      B. 82                      C. 113                      D. 122

2. 【2022】若三叉树  $T$  中有 244 个结点(叶结点的高度为 1), 则  $T$  的高度至少是( )。

【P124, 04 题】

- A. 8                      B. 7                      C. 6                      D. 5

### 4.2 二叉树基础

1. 【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( )。 【P1, 05 题】

- A. 39                      B. 52                      C. 111                      D. 119

2. 【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( )。 【P20, 04 题】

- A. 257                      B. 258                      C. 384                      D. 385

3. 【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( )。 【P20, 06 题】

- A. 115                      B. 116                      C. 1895                      D. 1896

4. 【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( )。 【P85, 04 题】

- A.  $2k - 1$                       B.  $2k$                       C.  $k^2$                       D.  $2^k - 1$

5. 【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( )。 【P105, 03 题】

- A. 31                      B. 16                      C. 15                      D. 10

6. 【2016】如果一棵非空 $k(k \geq 2)$ 叉树 $T$ 中每个非叶结点都有 $k$ 个孩子, 则称 $T$ 为正则 $k$ 叉树, 回答下列问题并给出推导过程: 【P73, 42 题】

(1) 若 $T$ 有 $m$ 个非叶结点, 则 $T$ 中的叶结点有多少个?

(2) 若 $T$ 的高度为 $h$ (单结点的树 $h = 1$ ), 则 $T$ 的结点数最多为多少个?最少为多少个?

### 4.3 二叉树的遍历

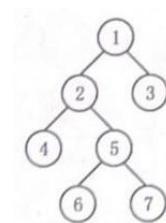
1. 【2009】给定二叉树如右图所示。设 $N$ 表示二叉树的根,  $L$ 表示根结点的左子树,  $R$ 表示根结点的右子树。若遍历后的结点序列是 3, 1, 7, 5, 6, 2, 4, 则其遍历方式是( )。 【P1, 03 题】

A. LRN

B. NRL

C. RLN

D. RNL



2. 【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( )。 【P20, 05 题】

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

3. 【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( )。 【P30, 03 题】

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

4. 【2015】先序序列为a,b,c,d的不同二叉树的个数是( )。 【P58, 02 题】

- A. 13                      B. 14                      C. 15                      D. 16

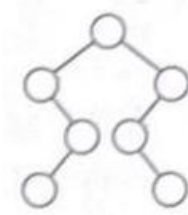
5. 【2017】要使一棵非空二叉树的先序序列与中序序列相同, 所有非叶结点须满足的条件是( )。 【P76, 04 题】

- 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

6. 【2017】已知一棵二叉树的树形如右图所示,其后序序列为 eacbdgf,树中与结点a同层的结点是 ( )。

【P76, 05 题】

- A. c                      B. d                      C. f                      D. g



7. 【2022】若结点p与q在二叉树T的中序遍历序列中相邻,且p在q之前,则下列p与q的关系中,不可能的是( )。

【P124, 03 题】

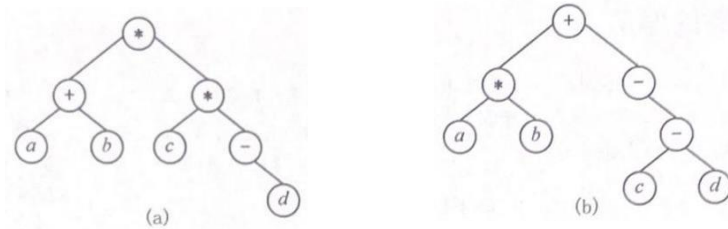
- I. q是p的双亲                      II. q是p的右孩子  
III. q是p的右兄弟                      IV. q是p的双亲的双亲

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

8. 【2017】设计算法, 将给定的表达式树(二叉树)转换为等价的中缀表达式(通过括号反映操作符的计算次序)并输出。例如, 当下列两棵表达式树作为输入时, 输出的等价中缀表达式分别为  $(a + b) * (c * (-d))$  和  $(a * b) + (-(c - d))$ 。

二叉树结点定义如下:

```
typedef struct node {  
    char data[10];        //存储操作数或符号数  
    struct node *left, *right;  
}BTree;
```



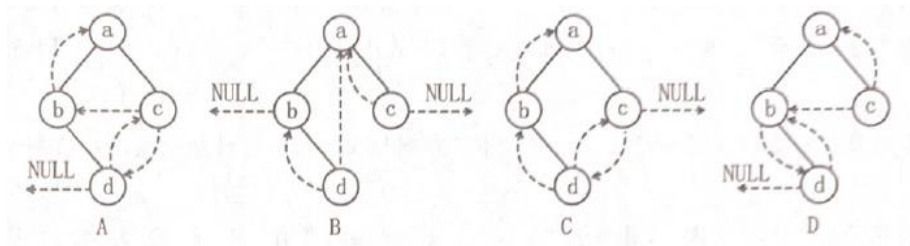
要求:

【P81, 41 题】

- (1) 给出算法的基本设计思想。
- (2) 根据设计思想, 采用C或C++语言描述算法, 关键之处给出注释。

## 4.4 线索二叉树

1. 【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( )。 【P11, 03 题】

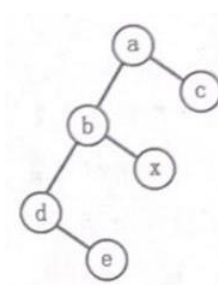


2. 【2013】 $X$ 是后序线索二叉树的叶结点,且 $X$ 存在左兄弟结点 $Y$ ,则 $X$ 右线索指向( ) 【P40, 05 题】

- 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

3. 【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.  $b$ 、 $a$



【P49, 04 题】

## 4.5 树与森林

1. 【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( )。 【P1, 06 题】

- 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

2. 【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( )。 【P49, 05 题】

- 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

3. 【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( )。 【P95, 02 题】

- A. 先序遍历      B. 中序遍历      C. 后序遍历      D. 按层遍历



4. 【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, 中根遍历序列是 badfec, 则 $T$ 的后根遍历序列是( )。 【P105, 04 题】

- A. badfec                      B. bdfeca                      C. bfedca                      D. fedcba

5. 【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( )。 【P115, 04 题】

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

## 4.6 树与二叉树的应用

1. 【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( )。 【P11, 06 题】

- 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

2. 【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( )。 【P39, 04 题】

- A. 27                      B. 46                      C. 54                      D. 56

3. 【2014】5 个字符有如下 4 种编码方案, 不是前缀编码的是( )。 【P49, 06 题】

- A. 01,0000,0001,001,1                      B. 011,000,001,010,1  
C. 000,001,010,011,100                      D. 0,100,110,1110,1100

5. 【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( )。 【P58, 03 题】

- 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

6. 【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( )。 【P77, 06 题】

- A. acgabfh                      B. *adbagbb*                      C. afbeagd                      D. afeefgd

6. 【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( )。 【P85, 05 题】

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

7. 【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( )。 【P95, 03 题】

A. 56

B. 57

C. 58

D. 60

8. 【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( )。 【P116, 05 题】

A. 89

B. 200

C. 208

D. 289

9. 【2022】对任意给定的含 $n(n > 2)$ 个字符的有限集 $S$ , 用二叉树表示 $S$ 的哈夫曼编码集和定长编码集, 分别得到二叉树  $T_1$  和  $T_2$ 。下列叙述中, 正确的是( )。 【P124, 05 题】

A.  $T_1$  与  $T_2$  的结点数相同

B.  $T_1$  的高度大于  $T_2$  的高度

C. 出现频次不同的字符在 $T_1$ 中处于不同的层

D. 出现频次不同的字符在 $T_2$ 中处于相同的层

10. 【2012】设有 6 个有序表 A、B、C、D、E、F, 分别含有 10、35、40、50、60 和 200 个数据元素, 各表中的元素按升序排列。要求通过 5 次两两合并, 将 6 个表最终合并成 1 个升序表, 并在最坏情况下比较的总次数最小。请回答下列问题: 【P35, 41 题】

(1) 给出完整的合并过程, 并求出最坏情况下比较的总次数。

(2) 根据你的合并过程, 描述 $N(N \geq 2)$ 个不等长升序表的合并策略, 并说明理由。

11. 【2014】二叉树的带权路径长度(WPL)是二叉树中所有叶结点的带权路径长度之和。给定一棵二叉树T,采用二叉链表存储,其结点结构为|left|weight|right|,其中叶结点的weight域保存该结点的非负权值。设root为指向T的根结点的指针,请设计求T的WPL的算法,要求:【P52,41题】

- (1) 给出算法的基本设计思想。
- (2) 使用C或C++语言,给出二叉树结点的数据类型定义。
- (3) 根据设计思想,采用C或C++语言描述算法,关键之处给出注释。

12. 【2020】若任一字符的编码都不是其他字符编码的前缀, 则称这种编码具有前缀特性, 现有某字符集 (字符个数  $\geq 2$ ) 的不等长编码, 每个字符的编码均为二进制的 0, 1 序列, 最长为  $L$  位, 且具有前缀特性。请回答下列问题: 【P111, 42 题】

- (1) 哪种数据结构适宜保存上述具有前缀特性的不等长编码?
- (2) 基于你所设计的数据结构, 简述从 0/1 串到字符串的译码过程。
- (3) 简述判定某字符集的不等长编码是否具有前缀特性的过程。

## 第 5 章图

### 5.1 图的基本概念

1. 【2009】下列关于无向连通图特性的叙述中, 正确的是( )。 【P2, 07 题】

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

2. 【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( )。 【P11, 07 题】

A. 6

B. 15

C. 16

D. 21

3. 【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( )。 【P77, 07 题】

A. 10

B. 11

C. 13

D. 15

4. 【2022】对于无向图 $G = (V, E)$ , 下列选项中, 正确的是( )。 【P125, 06 题】
- A. 当 $|V| > |E|$ 时,  $G$ 一定是连通的
- B. 当 $|V| < |E|$ 时,  $G$ 一定是连通的
- C. 当 $|V| = |E| - 1$ 时,  $G$ 一定是不连通的
- D. 当 $|V| > |E| + 1$ 时,  $G$ 一定是不连通的

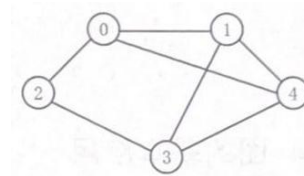
## 5.2 图的基本存储及基本操作

1. 【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( )。 【P40, 07 题】
- A. 1,2,1,2
- B. 2,2,1,1
- C. 3,4,2,3
- D. 4,4,2,2



2. 【2015】已知含有 5 个顶点的图 $G$ 如右图所示。

请回答下列问题：



【P63, 42 题】

(1) 写出图 $G$ 的邻接矩阵 $A$  (行、列的下标从 0 开始计算)。

(2) 求 $A^2$ , 矩阵 $A^2$ 中位于 0 行 3 列元素值的含义是什么?

(3) 若已知具有 $n$  ( $n \geq 2$ ) 个顶点的图的邻接矩阵为 $B$ , 则 $B^m$  ( $2 \leq m \leq n$ ) 中非零元素的含义是什么?

3. 【2021】已知无向连通图 $G$ 由顶点集 $V$ 和边集 $E$ 组成,  $|E| > 0$ , 当 $G$ 中度为奇数的顶点个数为不大于 2 的偶数时,  $G$ 存在包含所有边且长度为 $|E|$ 的路径(称为 EL 路径)。设图 $G$ 采用邻接矩阵存储, 类型定义如下:

```
typedef struct {           //图的定义
    int numVertices, numEdges; //图中实际的顶点数和边数
    char VerticesList[MAXV];   //顶点表。MAXV 为已定义常量
    int Edge[MAXV][MAXV];     //邻接矩阵
}MGraph;
```

请设计算法: `int IsExistEL(MGraph G)`, 判断 $G$ 是否存在 EL 路径, 若存在, 则返回 1, 否则返回 0。

要求:

【P121, 41 题】

- (1) 给出算法的基本设计思想。
- (2) 根据设计思想, 采用C或C++语言描述算法, 关键之处给出注释。
- (3) 说明你所设计算法的时间复杂度和空间复杂度。

### 5.3 图的遍历

1. 【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( )。 【P30, 05 题】

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

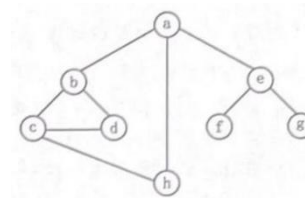
2. 【2013】 若对右图所示的无向图进行遍历, 则下列选项中, 不是广度优先遍历序列的是( )。 【P40, 08 题】

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



3. 【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( )。 【P58, 05 题】

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

4. 【2016】下列选项中,不是右图深度优先搜索序列的是( )。

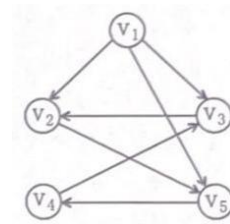
【P67, 06 题】

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.  $V_1, V_2, V_3, V_4, V_5$



## 5.4 图的基本应用

1. 【2010】对右图进行拓扑排序,可得不同拓扑排序的个数是( )。

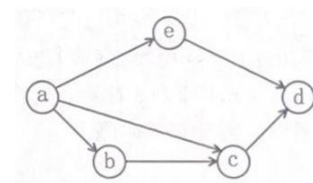
【P12, 08 题】

A. 4

B. 3

C. 2

D. 1



2. 【2011】下列关于图的叙述中,正确的是( )。

【P21, 08 题】

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

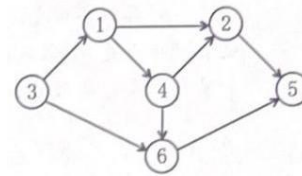
D. 仅 I、III

3. 【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( )。 【P30, 06 题】

- 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

4. 【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( )。 【P49, 07 题】

- 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



5. 【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( )。

【P67, 07 题】

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

6. 【2018】下列选项中不是右侧有向图的拓扑序列的是( )。

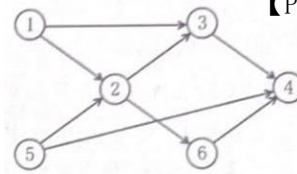
【P86, 07 题】

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



7. 【2021】给定右侧有向图, 该图的拓扑有序序列的个数是( )。

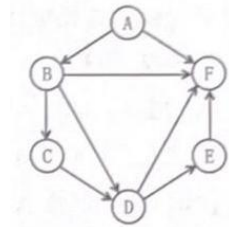
【P116, 07 题】

A. 1

B. 2

C. 3

D. 4



8. 【2012】下列关于最小生成树的陈述中, 正确的是( )。

【P31, 08 题】

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

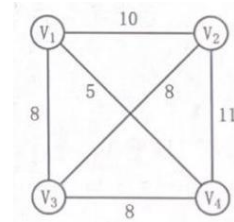
9. 【2015】求右侧带权图的最小(代价)生成树时,可能是 Kruskal 算法第 2 次选中但不是 Prim 算法(从 $V_4$ 开始)第 2 次选中的边是( )。

A.  $(V_1, V_3)$

B.  $(V_1, V_4)$

C.  $(V_2, V_3)$

D.  $(V_3, V_4)$



【P58, 06 题】

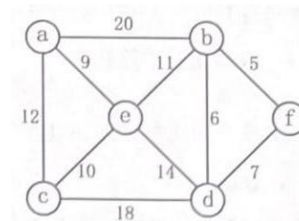
10. 【2020】已知无向图 $G$ 如右所示,使用克鲁斯卡尔(Kruskal)算法求图 $G$ 的最小生成树,加到最小生成树中的边依次是( )。

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.  $(a, e), (c, e), (b, e), (b, f), (b, d)$



【P106, 07 题】

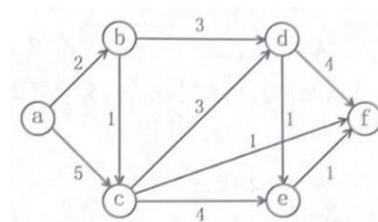
11. 【2012】对右图所示的有向带权图,若采用 Dijkstra 算法求从源点 $a$ 到其他各顶点的最短路径,则得到的第一条最短路径的目标顶点是 $b$ ,第二条最短路径的目标顶点是 $c$ ,后续得到的其余各最短路径的目标顶点依次是( )。

A.  $d, e, f$

B.  $e, d, f$

C.  $f, d, e$

D.  $f, e, d$



【P30, 07 题】

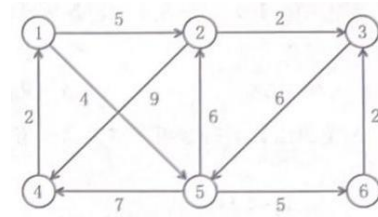
12. 【2016】使用 Dijkstra 算法求右图中从顶点 1 到其他各顶点的最短路径，依次得到的各最短路径的目标顶点是( )。

A. 5,2,3,4,6

B. 5,2,3,6,4

C. 5,2,4,3,6

D. 5,2,6,3,4



【P68, 08 题】

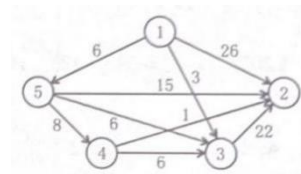
13. 【2021】使用 Dijkstra 算法求右图中从顶点 1 到其余各顶点的最短路径, 将当前找到的从顶点 1 到顶点 2, 3, 4, 5 的最短路径长度保存在数组 dist 中, 求出第二条最短路径后, dist 内容更新为( )。

A. 26,3,14,6

B. 25,3,14,6

C. 21,3,14,6

D. 15,3,14,6



【P116, 08 题】

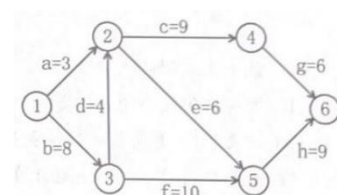
14. 【2013】右侧的 AOE 网表示一项包含 8 个活动的工程。通过同时加快若干活动的进度可以缩短整个工程的工期。下列选项中，加快其进度就可以缩短工期工程的是( )。

A. *c* 和 *e*

B. *d* 和 *c*

C. *f* 和 *d*

D. *f* 和 *h*



【P40, 09 题】



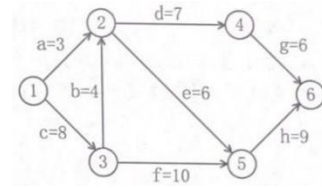
15. 【2019】如右图所示的 AOE 网表示一项包含 8 个活动的工程。活动  $d$  的最早开始时间和最迟开始时间分别是( )。

A. 3 和 7

B. 12 和 12

C. 12 和 14

D. 15 和 15



【P96, 05 题】

16. 【2020】若使用 AOE 网估算工程进度, 则下列叙述中正确的是( )。

【P106, 08 题】

- 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

17. 【2022】右图是一个有 10 个活动的 AOE 网, 时间余量最大的活动是( )。

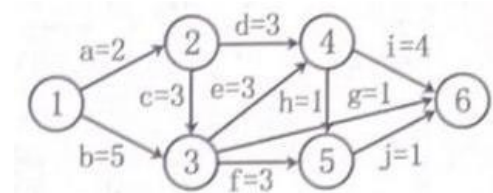
【P125, 07 题】

A. c

B. g

C. h

D. j



18. 【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( )

【P96, 06 题】

- A. 5                      B. 6                      C. 8                      D. 9

19. 【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( )。

【P105, 06 题】

- 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

20. 【2009】带权图（权值非负，表示边连接的两顶点间的距离）的最短路径问题是找出从初始顶点到目标顶点之间的一条最短路径，假设从初始顶点到目标顶点之间存在路径，现有一种解决该问题的方法：

【P6, 41 题】

- (1) 设最短路径初始时仅包含初始顶点，令当前顶点 $u$ 为初始顶点。
- (2) 选择离 $u$ 最近且尚未在最短路径中的一个顶点 $v$ ，加入最短路径，修改当前顶点 $u = v$ 。
- (3) 重复步骤(2)，直到 $u$ 是目标顶点时为止。

请问上述方法能否求得最短路径？若该方法可行，请证明；否则，请举例说明。

21. 【2011】 已知有 6 个顶点(顶点编号为0 ~ 5)的有向带权图 $G$ , 其邻接矩阵 $A$ 为上三角矩阵, 按行为主序(行优先)保存在如下的一维数组中。

4	6	$\infty$	$\infty$	$\infty$	5	$\infty$	$\infty$	$\infty$	4	3	$\infty$	$\infty$	3	3
---	---	----------	----------	----------	---	----------	----------	----------	---	---	----------	----------	---	---

要求：

【P24, 41 题】

- (1) 写出图 $G$ 的邻接矩阵 $A$ 。
- (2) 画出有向带权图 $G$ 。
- (3) 求图 $G$ 的关键路径, 并计算该关键路径的长度。

22. 【2014】某网络中的路由器运行 OSPF 路由协议,表 5.1 是路由器 R1 维护的主要链路状态信息 (LSI),R1 构造的网络拓扑图如图 5.1 所示,是根据下表及 R1 的接口名构造出来的网络拓扑。

请回答下列问题: 【P53, 42 题】

- (1) 本题中的网络可抽象为数据结构中的哪种逻辑结构?
- (2) 针对表中的内容,设计合理的链式存储结构,以保存表中的链路状态信息 (LSI)。要求给出链式存储结构的数据类型定义,并画出对应表的链式存储结构示意图(示意图中可仅以 ID 标识结点)。
- (3) 按照 Dijkstra 算法的策略,依次给出 R1 到达子网 192.1.x.x 的最短路径及费用。

表 5.1-R1 所维护的 LSI

		R1 的 LSI	R2 的 LSI	R3 的 LSI	R4 的 LSI	备注
RouterID		10.1.1.1	10.1.1.2	10.1.1.5	10.1.1.6	标识路由器的 IP 地址
Link1	ID	10.1.1.2	10.1.1.1	10.1.1.6	10.1.1.5	所连路由器的 RouterID
	IP	10.1.1.1	10.1.1.2	10.1.1.5	10.1.1.6	Link1 的本地 IP 地址
	Metric	3	3	6	6	Link1 的费用
Link2	ID	10.1.1.5	10.1.1.6	10.1.1.1	10.1.1.2	所连路由器的 RouterID
	IP	10.1.1.9	10.1.1.13	10.1.1.10	10.1.1.14	Link2 的本地 IP 地址
	Metric	2	4	2	4	Link2 的费用
Net1	Prefix	192.1.1.0/	192.1.6.0/2	192.1.5.0/	192.1.7.0/2	直连网络 Net1 的网络前缀
	Metric	1	1	1	1	到达直连网络 Net1 的费用

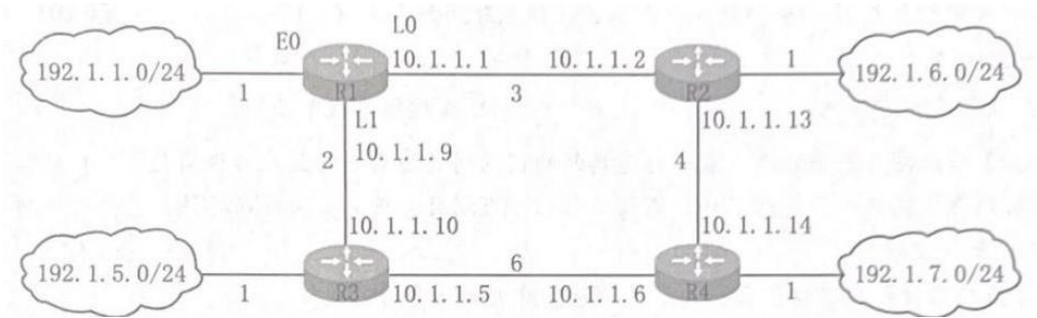


图 5.1 所构造的网络拓扑

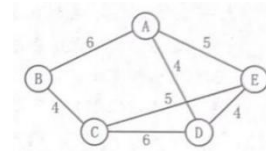
23. 【2017】使用 Prim 算法求带权连通图的最小(代价)生成树(MST)。请回答下列问题：

(1) 右图 $G$ , 从顶点 $A$ 开始求 $G$ 的 MST, 依次给出按算法选出的边。

【P81, 42 题】

(2) 图 $G$ 的 MST 是唯一的吗?

(3) 对任意的带权连通图, 满足什么条件时, 其 MST 是唯一的?

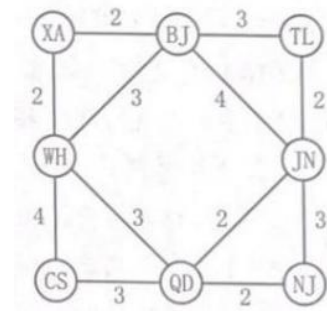


24. 【2018】拟建设一个光通信骨干网络连通 BJ, CS, XA, QD, JN, NJ, TL 和 WH 等 8 个城市, 右图中无向边上的权值表示两个城市之间备选光缆的铺设费用。请回答下列问题: 【P90, 42 题】

(1) 仅从铺设费用角度出发, 给出所有可能的最经济的光缆铺设方案(用带权图表示), 并计算相应方案的总费用。

(2) 该图可采用图的哪种存储结构? 给出求解问题(1)所用的算法名称。

(3) 假设每个城市采用一个路由器按(1)中得到的最经济方案组网, 主机H1直接连接在TL的路由器上, 主机H2直接连接在 BJ 的路由器上。若H1向H2发送一个 TTL= 5 的 IP 分组, 则H2是否可以收到该 IP 分组?



## 第 6 章查找

### 6.1 顺序查找、折半查找与分块查找

1. 【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( )。 【P12, 09 题】

- A. 4                      B. 5                      C. 6                      D. 7

2. 【2015】下列选项中不能构成折半查找中关键字比较序列的是( )。 【P59, 07 题】

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

3. 【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:

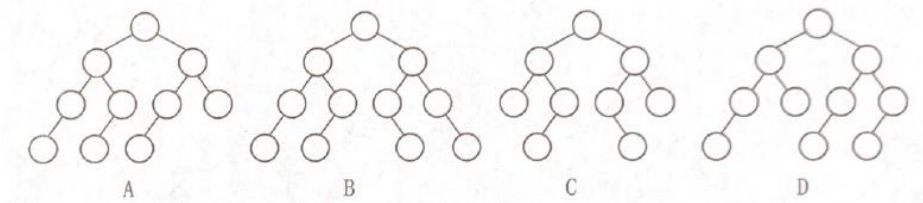
```
k = 0;
while ( k < n 且 A[k] < x )    k = k + 3;
if (k < n 且 A[k] == x)      查找成功;
elseif(k-1 < n 且 A[k-1] == x) 查找成功;
elseif(k-2 < n 且 A[k-2] == x) 查找成功;
else 查找失败;
```

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( )。 【P68, 09 题】

- 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



4. 【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（ ）。【P77, 08 题】



5. 【2013】设包含 4 个数据元素的集合  $S = \{ "do", "for", "repeat", "while" \}$ , 各元素的查找概率依次为:  $p_1 = 0.35, p_2 = 0.15, p_3 = 0.15, p_4 = 0.35$ 。将  $S$  保存在一个长度为 4 的顺序表中, 采用折半查找法, 查找成功时的平均查找长度为 2.2。请回答下列问题: 【P45, 42 题】

- (1) 若采用顺序存储结构保存  $S$ , 且要求平均查找长度更短, 则元素应如何排列? 应使用何种查找方法? 查找成功时的平均查找长度是多少?
- (2) 若采用链式存储结构保存  $S$ , 且要求平均查找长度更短, 则元素应如何排列? 应使用何种查找方法? 查找成功时的平均查找长度是多少?

## 6.2 二叉搜索树、平衡二叉树和红黑树

1. 【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( )。

【P21, 07 题】

- 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

2. 【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( )。

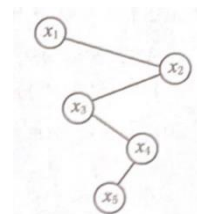
【P40, 06 题】

- 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

3. 【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( )。

【P86, 06 题】

- A.  $x_1 < x_2 < x_5$   
B.  $x_1 < x_4 < x_5$   
C.  $x_3 < x_5 < x_4$   
D.  $x_4 < x_3 < x_5$



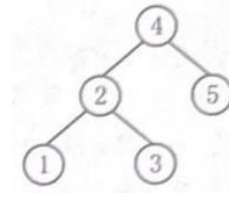
4. 【2020】下列给定的关键字输入序列中,不能生成如右图所示二叉排序树的是( )。

A. 4,5,2,1,3

B. 4,5,1,2,3

C. 4,2,5,3,1

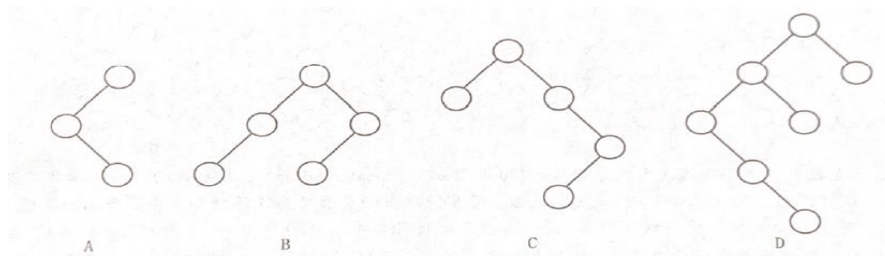
D. 4,2,1,3,5



【P105, 05 题】

5. 【2009】下列二叉排序树中,满足平衡二叉树定义的是( )。

【P1, 04 题】



6. 【2010】在右图所示的平衡二叉树中,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( )。

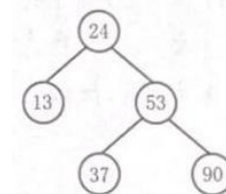
【P11, 04 题】

A. 13, 48

B. 24, 48

C. 24, 53

D. 24, 90



7. 【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( )。 【P30, 04 题】

- A. 12                      B. 20                      C. 32                      D. 33

8. 【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( )。 【P39, 03 题】

- A. 0                      B. 1                      C. 2                      D. 3

9. 【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( )。 【P58, 04 题】

- 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. 树中最大元素一定是无左子树

10. 【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( )。 【P95, 04 题】

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

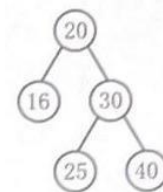
11. 【2021】给定平衡二叉树如右图所示, 插入关键字 23 后, 根中的关键字是( ) 【P116, 06 题】

A. 16

B. 20

C. 23

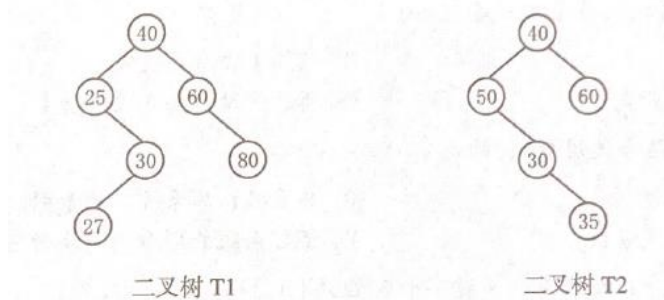
D. 25



12. 【2022】已知非空二叉树T的结点值均为正整数,采用顺序存储方式保存,数据结构定义如下:

```
typedef struct{                                //MAX_SIZE 为已定义常量
    int SqBiTNode[MAX_SIZE];                //保存二叉树结点值的数组
    int ElemNum;                             //实际占用的数组元素个数
}SqBiTree;
```

T中不存在的结点在数组 SqBiTNode 中用-1 表示。例如,对于下图所示的两棵非空二叉树 T1 和 T2:



T1 的存储结果如下:

40	25	60	-1	30	-1	80	-1	-1	27		
----	----	----	----	----	----	----	----	----	----	--	--

T1. SqBiTNode

T1. ElemNum= 10

T2 的存储结果如下:

40	50	60	-1	30	-1	-1	-1	-1	-1	35	
----	----	----	----	----	----	----	----	----	----	----	--

T2. SqBiTNode

T2. ElemNum= 11

请设计一个尽可能高效的算法,判定一棵采用这种方式存储的二叉树是否为二叉搜索树,若是,则返回 true,否则,返回 false。要求: 【P130, 41 题】

(1) 给出算法的基本设计思想。

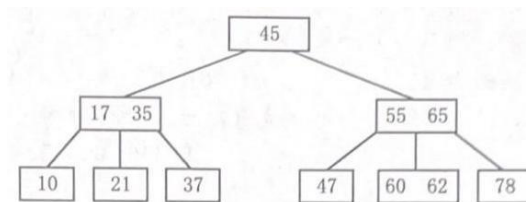
(2) 根据设计思想,采用C或C++语言描述算法,关键之处给出注释。

### 6.3 B、B<sup>+</sup>树基础

1. 【2009】以下叙述中, 不符合 $m$ 阶B树定义要求的是( )。 【P2, 08 题】

- 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

2. 【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( )。 【P31, 09 题】



- A. 60
- B. 60, 62
- C. 62, 65
- D. 65

3. 【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( )。 【P41, 10 题】

- A. 5
- B. 7
- C. 8
- D. 14

4. 【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( )。 【P49, 09 题】

- A. 5                                  B. 6                                  C. 10                                  D. 15

5. 【2016】B<sup>+</sup>树不同于B树的特点之一是( )。 【P68, 10 题】

- 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

6. 【2017】下列应用中, 适合使用B<sup>+</sup>树的是( )。 【P77, 09 题】

- 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

7. 【2018】高度为 5 的 3 阶 B 树含有的关键字个数至少是( )。 【P86, 08 题】

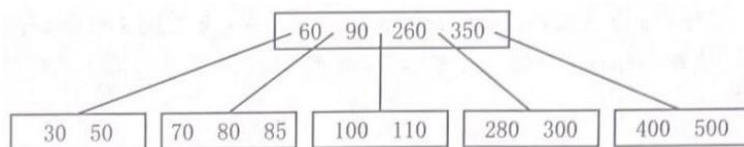
- A. 15                                  B. 31                                  C. 62                                  D. 242



8. 【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( )。 【P106, 10 题】
- A. 8                      B. 6, 9                      C. 8,13                      D. 9, 12

9. 【2021】在一棵高度为3、阶数为3的B树中,根为第1层,若第2层个有4个关键字,则该树的结点个数最多是( )。 【P116, 09 题】
- A. 11                      B. 10                      C. 9                      D. 8

10. 【2022】在下图所示的5阶B树 $T$ 中,删除关键字260之后需要进行必要的调整,得到新的B树 $T_1$ 。下列选项中,不可能是 $T_1$ 根结点中关键字序列的是( )。 【P125, 08 题】



- A. 60,90,280                      B. 60,90,350                      C. 60,85,110,350                      D. 60,90,110,350

## 6.4 散列(Hash)表

1. 【2011】为提高散列(Hash)表的查找效率,可以采取的正确措施是( )。 【P21, 09 题】
- 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

2. 【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( )。 【P49, 08 题】

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

3. 【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 $HT$ 后,查找成功的平均查找长度是( )。 【P86, 09 题】

- A. 1.5                          B. 1.6                          C. 2                                  D. 3

4. 【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( )。 【P97, 08 题】

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

5. 【2022】下列因素中,影响散列(哈希)方法平均查找长度的是( )。 【P125, 09 题】

- I. 装填因子                  II. 散列函数                  III. 冲突解决策略
- A. 仅 I、II                  B. 仅 I、III                  C. 仅 II、III                  D. I、II、III

6. 【2010】将关键字序列(7,8,30,11,18,9,14)散列存储到散列表中。散列表的存储空间是一个下标从 0 开始的一维数组,散列函数为: $H(\text{key}) = (\text{key} \times 3) \bmod 7$ ,处理冲突采用线性探测再散列法,要求装填(载)因子为 0.7。请回答下列问题。 【P16, 41 题】

(1)请画出所构造的散列表。

(2)分别计算等概率情况下查找成功和查找不成功的平均查找长度。

## 6.5 字符串匹配模式

1. 【2015 年】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 ( )。

【P59, 08 题】

- A.  $i = 1, j = 0$       B.  $i = 5, j = 0$       C.  $i = 5, j = 2$       D.  $i = 6, j = 2$

2. 【2019 年】设主串  $T = \text{"abaabaabcabaabc"}$ , 模式串  $S = \text{"abaabc"}$ , 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 ( )。

【P97, 09 题】

- A. 9      B. 10      C. 12      D. 15

## 第 7 章排序

### 7.1 插入类排序

1. 【2012】对同一待排序序列, 分别进行折半插入排序和直接插入排序, 两者之间可能的不同之处是( )。 【P31, 11 题】

- 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

2. 【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( ) 【P49, 10 题】

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

3. 【2015】希尔排序的组内排序采用的是( )。 【P59, 11 题】

- A. 直接插入排序
- B. 折半插入排序
- C. 快速排序
- D. 归并排序

4. 【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( )。 【P86, 10 题】

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

## 7.2 交换类排序

1. 【2010】采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( )。

【P12, 10 题】

- 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

2. 【2011】为实现快速排序算法, 待排序序列宜采用的存储方式是( )。

【P21, 10 题】

- A. 顺序存储
- B. 散列存储
- C. 链式存储
- D. 索引存储

3. 【2014】下列选项中, 不可能是快速排序第 2 趟排序结果的是( )。

【P49, 11 题】

- 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

4. 【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( )。 【P97, 10 题】

- 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

5. 【2016】已知由 $n(n \geq 2)$ 个正整数构成的集合 $A = \{a_k \mid 0 \leq k < n\}$ , 将其划分为两个不相交的子集 $A_1$ 和 $A_2$ , 元素个数分别是 $n_1$ 和 $n_2$ ,  $A_1$ 和 $A_2$ 中元素之和分别为 $S_1$ 和 $S_2$ 。设计一个尽可能高效的划分算法, 满足 $|n_1 - n_2|$ 最小且 $|S_1 - S_2|$ 最大。要求: 【P73, 43 题】

- (1) 给出算法的基本设计思想;
- (2) 根据设计思想, 采用C或C++语言描述算法, 关键之处给出注释;
- (3) 说明你所设计算法的时间复杂度和空间复杂度。

### 7.3 选择类排序

1. 【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( )。 【P2, 09 题】

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

2. 【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( )。 【P21, 11 题】

A. 1

B. 2

C. 4

D. 5

3. 【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( )。 【P59, 10 题】

A. 1

B. 2

C. 3

D. 4

4. 【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( )。 【P86, 11 题】

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

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

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

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



5. 【2020】 以下关于大根堆(至少含 2 个元素)的叙述中正确的是( )。 【P106, 09 题】

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

6. 【2021】 将关键字 6,9,1,5,8,4,7 依次插入初始为空的大根堆 H, 得到的 H 是( )。 【P117, 11 题】

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

7. 【2022】现有 $n(n > 100000)$ 个数保存在一维数组 $M$ 中, 需要查找 $M$ 中最小的 10 个数。请回答下列问题: 【P131, 42 题】

(1) 设计一个完成上述查找任务的算法, 要求平均情况下的比较次数尽可能少, 简述其算法思想(不需要程序实现)。

(2) 说明你所设计的算法平均情况下的时间复杂度和空间复杂度。

## 7.4 二路归并排序和基数排序

1. 【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( )。 【P41, 11 题】

- 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

2. 【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( )。

【P77, 10 题】

- 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

3. 【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( )。

【P117, 10 题】

- A. 43,892                      B. 236, 301                      C. 301, 892                      D. 485, 301

4. 【2022】使用二路归并排序对含 $n$ 个元素的数组 $M$ 进行排序时,二路归并操作的功能是( )。

A. 将两个有序表合并为一个新的有序表 【P126, 10 题】

B. 将 $M$ 划分为两部分,两部分的元素个数大致相等

C. 将 $M$ 划分为 $n$ 个部分,每个部分中仅含有一个元素

D. 将 $M$ 划分为两部分,一部分元素的值均小于另一部分元素的值

## 7.5 各种内部排序算法总结

1. 【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( )。 【P2, 10 题】

A. 冒泡排序

B. 插入排序

C. 选择排序

D. 二路归并排序

2. 【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( )。 【P12, 11 题】

A. 冒泡排序

B. 希尔排序

C. 归并排序

D. 基数排序

3. 【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是( )。 【P31, 10 题】

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

4. 【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( )。 【P59, 09 题】

A. 直接插入排序

B. 冒泡排序

C. 基数排序

D. 快速排序

5. 【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( )。

【P77, 11 题】

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

6. 【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

【P97, 07 题】

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

7. 【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( )。 【P106, 11 题】

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. 【2022】对数据进行排序时,若采用直接插入排序而不是快速排序,则可能的原因是( )。

【P126, 11 题】

I. 大部分元素已有序

II. 待排序的元素数量很少

III. 要求空间复杂度为 $O(1)$

IV. 要求排序算法是稳定的

A. 仅 I、II

B. 仅 III、IV

C. 仅 I、II、IV

D. I、II、III、IV

9. 【2021】已知某排序算法如下：

```
void cmpCountSort(int a[],int b[],int n) {  
    int i, j , *count;  
    count=(int *)malloc(sizeof(int) *n);           //C++语言:count=new int [n];  
    for(i=0; i<n; i++) count[i]=0;  
    for(i=0; i<n-1; i++)  
        for(j=i+1; j<n; j++)  
            if(a[i] < a[j]) count[j]++;  
            else count[i]++;  
    for(i=0;i<n;i++)b[count[i]]=a[i];  
    free(count);                                   //C++语言:deletecount;
```

请回答下列问题：

【P121, 42 题】

- (1) 若有  $\text{int } a[7] = \{25, -10, 25, 10, 11, 19\}$ ,  $b[6]$ , 则调用  $\text{cmpCountSort}(a, b, 6)$  后数组  $b$  中的内容是什么？
- (2) 若  $a$  中含有  $n$  个元素, 则算法执行过程中, 元素之间的比较次数是多少？
- (3) 该算法是稳定的吗?若是, 则阐述理由, 否则, 修改为稳定排序算法。

## 7.6 外部排序

1. 【2016】对 10 TB 的数据文件进行排序, 应使用的方法是( )。 【P68, 11 题】

- A. 希尔排序                      B. 堆排序                      C. 快速排序                      D. 归并排序

2. 【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( )。 【P98, 11 题】

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