

2009-2020 计算机考研 408 真题

数据结构部分

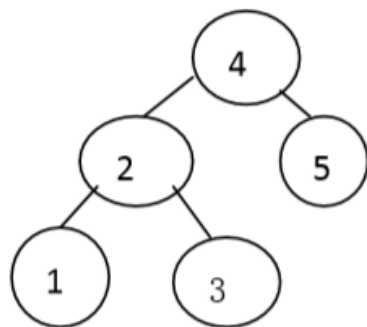
刘童鞋 2021 年 6 月 整理

注：文中答案来源网络仅供参考。

2020 年统考 408 真题-数据结构部分

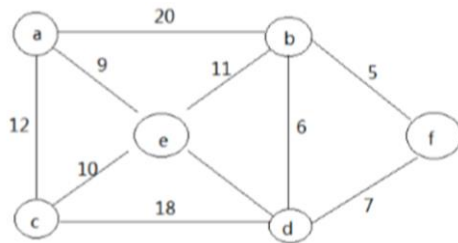
数据结构部分选择题:

1. 将一个 $10 * 10$ 对称矩阵 M 的上三角部分的元素 m_{ij} ($1 \leq i \leq j \leq 10$) 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
2. 对空栈 S 进行 Push 与 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
3. 对于任意一棵高度为 5 且有 10 个节点的二叉树, 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), 则存放该二叉树需要的存储单元数量至少是 ()。
A、31 B、16 C、15 D、10
4. 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 a, b, c, d, e, f , 中根遍历序列是 b, a, d, f, e, c 则 T 的后根遍历序列是 ()。
A、 b, a, d, f, e, c
B、 b, d, f, e, c, a
C、 b, f, e, d, c, a
D、 f, e, d, c, b, a
5. 下列给定的关键字输入序列中, 不能生成如下二叉排序树的是 ()。



6. 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) 定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

7. 已知无向图 G 如下所示，使用克鲁斯卡尔（Kruskal）算法求图 G 的最小生成树，加入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)(e,c)
 C、(a,e)(b,e)(c,e)(b,d)(b,f)
 D、(a,e)(c,e)(b,e)(b,f)(b,d)
8. 若使用 AOE 网估算工程进度，则下列叙述中正确的是（ ）。
- A、关键路径是从原点到汇点边数最多的一条路径
 B、关键路径是从原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
9. 下列关于大根堆（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。
10. 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，根节点中包含的关键字是（ ）。
- A、8 B、6, 9 C、8, 13 D、9, 12
11. 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

数据结构部分大题

12. 定义三元组 (a, b, c) (a, b, c 均为正数) 的距离 $D = |a-b| + |b-c| + |c-a|$. 给定 3 个非空整数集合 S_1, S_2, S_3 , 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) ($a \in S_1, b \in S_2, c \in S_3$) 中的最小距离。

例如 $S_1 = \{-1, 0, 9\}, S_2 = \{-25, -10, 10, 11\}, S_3 = \{2, 9, 17, 30, 41\}$ 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;
- (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。

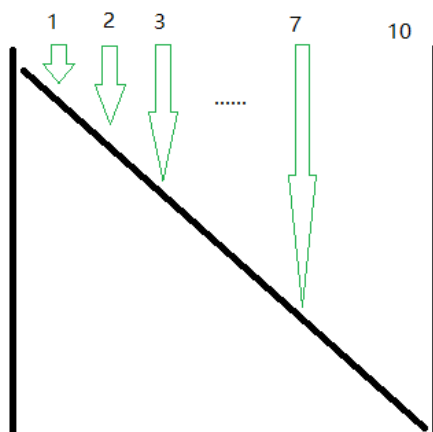
13. 若任一个字符的编码都不是其他字符编码的前缀, 则称这种编码具有前缀特性。现有某字符集 (字符个数 ≥ 2) 的不等长编码, 每个字符的编码均为二进制的 0, 1 序列, 最长为 L 位, 且具有前缀特性。请回答下列问题:

- (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

1. 【皮皮答案】C

【皮皮解析】

上三角矩阵列优先的存储模式：先存储具有一个元素第一列，再存储具有两个元素的第二列，以此类推



$m_{7,2}$ 说明 1-6 列均已存满，，故此元素是第 $1+2+3+4+5+6+2$ 个被存储单元

【注意!】C 语言数组的下标从 0 开始

故 $m_{7,2}$ 在 N 中的下标为 $23-1=22$ ，即 $N[22]$

2. 【皮皮答案】D

【皮皮解析】

操作	执行该操作后的栈(左侧为栈底)	出栈元素
Push	a	
Push	ab	
Pop	a	b
Push	ac	
Pop	a	c
Push	ad	
Push	ade	
Pop	ad	e

3. 【皮皮答案】A

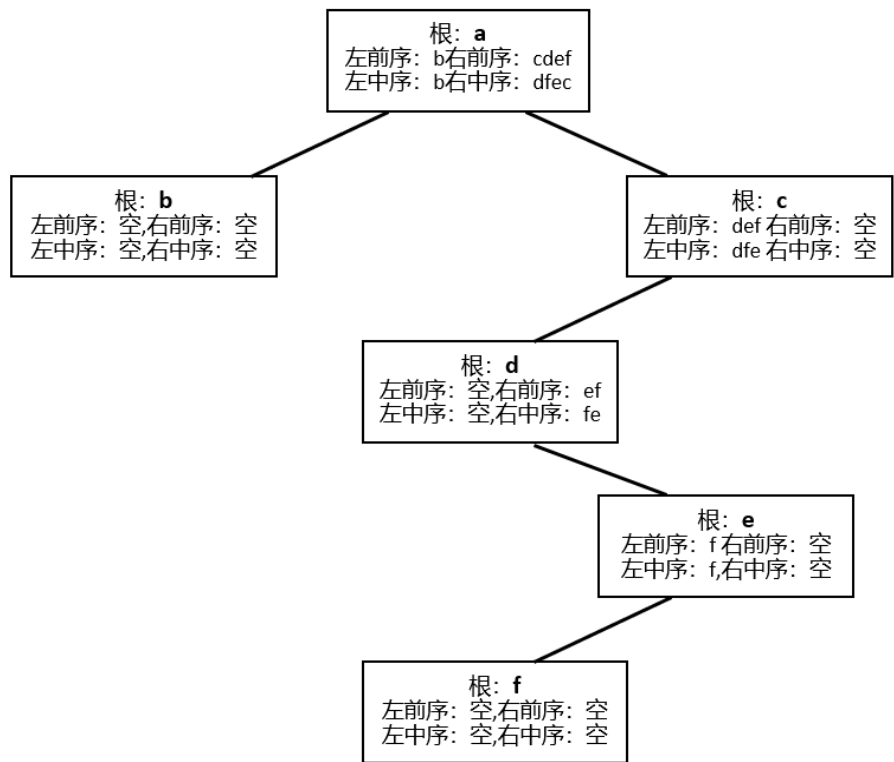
【皮皮解析】

本二叉树使用顺序结构存储时，为了保证任意性，其 1-5 层的所有节点（包括空节点）全部都要被存储起来，即考虑成一棵 5 层的满二叉树，存储单元大小为 $1+2+4+8+16 = 31$

4. 【皮皮答案】C

【皮皮解析】

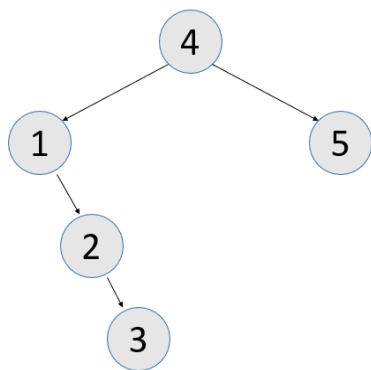
任何 n 个不同节点的二叉树，都可由它的中序序列和先序序列唯一确定。



此二叉树 T 对应的后根遍历序列是 bfedca

5. 【皮皮答案】B

【皮皮解析】基本概念题。



B 选项构造出的二叉排序树

6. 【皮皮答案】B

【皮皮解析】

DFS 是一个递归算法,在遍历的过程中,先访问的点被压入栈底.拓扑有序是指如果点 U 到点 V 有一条弧,则在拓扑序列中 U 一定在 V 之前.深度优先算法搜索路径恰恰是一条弧,栈的输出是从最后一个被访问点开始输出,最后一个输出的点是第一个被访问的点.所以是逆拓扑有序序列

7. 【皮皮答案】A

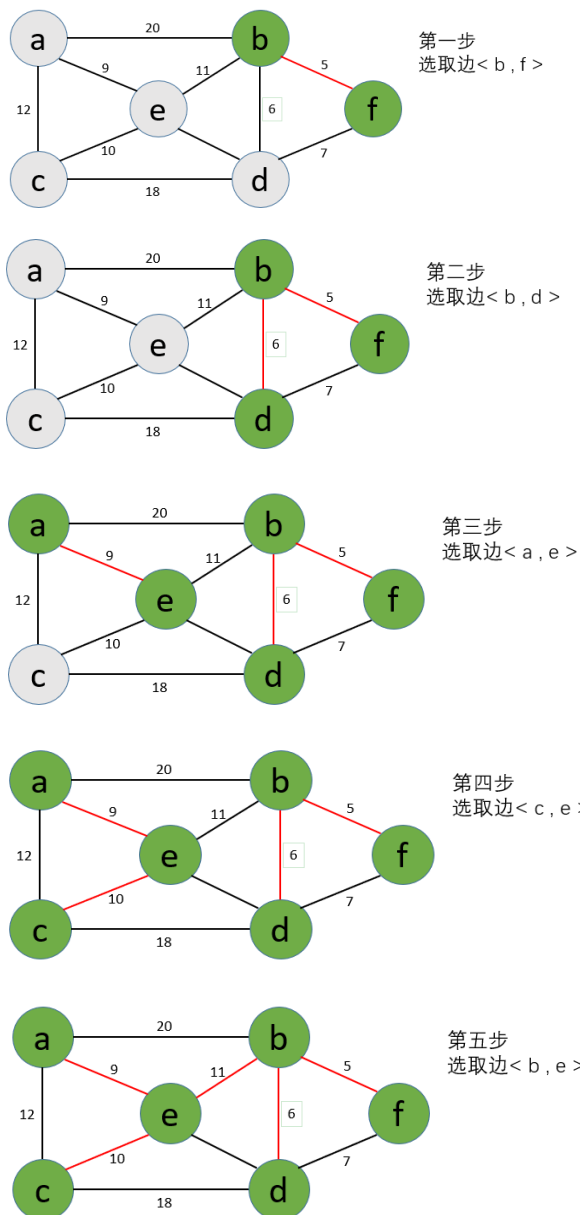
【皮皮解析】

Kruskal 算法：按权值递增次序选择合适的边构造最小生成树

基本思想：按照权值的递增顺序选择 $n-1$ 条边，并保证这 $n-1$ 条边不构成回路。

具体做法：首先构造一个只含 n 个顶点的森林，然后依权值从小到大从连通网中选择边加入到森林中，并使森林中不产生回路，直至森林变成一棵树为止。

*注：de 边权值尚不明确，但一定是比 11 大的某个数，不影响解题



第 1 步：选取边 $\langle b, f \rangle$

边 $\langle b, f \rangle$ 的权值最小，因此将它加入到最小生成树中。

第 2 步：选取边 $\langle b, d \rangle$

上一步操作之后，边 $\langle b, d \rangle$ 的权值最小，将它加入到最小生成树结果中。

第 3 步：选取边 $\langle a, e \rangle$

上一步操作之后，边 $\langle d, f \rangle$ 的权值最小，但 $\langle d, f \rangle$ 会和已有的边构成回路；因此，跳过边 $\langle d, f \rangle$ ，将边 $\langle a, e \rangle$ 它加入到最小生成树中。

第 4 步：选取边<c, e>

上一步操作之后，边<c, e>的权值最小，将它加入到最小生成树中。

第 5 步：选取边<b, e>

上一步操作之后，边<b, e>的权值最小，将它加入到最小生成树中。

此时，最小生成树构造完成, 它包括的边是：<b, f> <b, d> <a, e> <c, e> <b, e>

8. 【皮皮答案】B

【皮皮解析】基础概念题。

关键路径（critical path）：在 AOE 网中，从源点到汇点的所有路径中具有最大路径长度的路径

9. 【皮皮答案】I、 II、 IV

【皮皮解析】基础概念题

堆（Heap）具有以下特点：

- 1) 完全二叉树
- 2) 存储的值是偏序

大根堆（Max-heap）：父节点的值大于或等于子节点的值

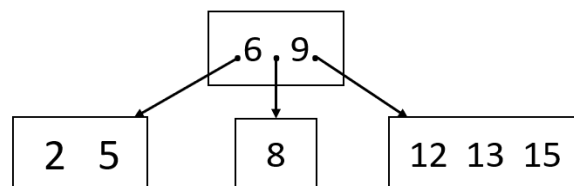
一般用数组（顺序结构）来表示堆

10. 【皮皮答案】B

【皮皮解析】

B 树是一种平衡的多路搜索树， 结点最大的孩子数目称为 B 树的阶。一个 m 阶 B 树具有如下属性：

- 1.定义任意非叶子结点最多只有 M 个儿子；且 $M > 2$
- 2.根结点的儿子数为 $[2, M]$
- 3.除根结点以外的非叶子结点的儿子数为 $[M/2, M]$
- 4.每个结点存放至少 $M/2 - 1$ （取上整）和至多 $M - 1$ 个关键字；（至少 2 个关键字）
- 5.非叶子结点的关键字个数 = 指向儿子的指针个数-1
- 6.非叶子结点的关键字： $K[1], K[2], \dots, K[M-1]$ ；且 $K[i] < K[i+1]$
- 7.非叶子结点的指针： $P[1], P[2], \dots, P[M]$ ；其中 $P[1]$ 指向关键字小于 $K[1]$ 的子树， $P[M]$ 指向关键字大于 $K[M-1]$ 的子树， 其它 $P[i]$ 指向关键字属于 $(K[i-1], K[i])$ 的子树
- 8.所有叶子结点位于同一层



最后生成的 B 树

11. 【皮皮答案】A

【皮皮解析】基础概念题

直接插入排序与简单选择排序相比

- 1) 直接插入排序元素比较次数少
- 2) 简单选择排序移动的元素少
- 3) 均为就地排序，空间复杂度均为 $O(1)$

12. 【皮皮解析】

方法一【暴力法】

三层 for 循环，时间复杂度为 $O(l*m*n)$ ， l, m, n 分别为集合 S_1, S_2, S_3 的长度

评分标准：7 分

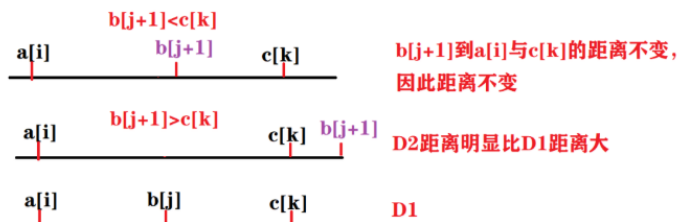
方法二

假设当前遍历到的这三个数组中的元素分别为 $a[i], b[j], c[k]$ ，并且有 $a[i] \leq b[j] \leq c[k]$ ，则最小距离为 $D1 = |a[i] - b[j]| + |b[j] - c[k]| + |c[k] - a[i]|$ ，那么接下来有三种情况：

1. 接下来求 $a[i], b[j], c[k+1]$ 的最小距离，因为 $c[k+1] > c[k]$ ，所以，此时的最小距离为 $D2$ ，肯定大于 $D1$



2. 接下来求 $a[i], b[j+1], c[k]$ 的最小距离，如果 $b[j+1] \leq c[k]$ ，则最小距离不变，如果 $b[j+1] > c[k]$ ，此时的最小距离为 $D2$ 肯定大于 $D1$



3. 接下来求 $a[i+1], b[j], c[k]$ 的最小距离，在该情况下 D 才有可能减小！

【因此只需移动最小的元素 $a[i]$ ！】

所以，整体的思路是开始得出三个数组第一个元素的最小距离，接下来移动最小三个元素中最小元素的下标，与之前得到的最小距离比较，看是否需要更新最小距离，直到遍历完三个数组

【时间复杂度为 $O(l+m+n)$ 】

13. 【皮皮解析】考点：前缀编码，赫夫曼(Huffman)树

(1) 二叉树或赫夫曼树

普通的二叉树也可以设计前缀编码，赫夫曼树会使总长最小，是对前者的优化。

(2) 将所有的字符信息存储到二叉树的叶子结点上，且约定左分支表示字符'0'，右分支表示字符'1'，则可将根节点到叶子结点的路径上分支字符组成的字符串作为该叶子结点字符的编码。从根节点出发将 0/1 串沿着分支探查下去，遇到带有信息的叶子结点即为一个字符，然后再从根节点出发，以此类推直至 0/1 串全部译码为字符串。

(3) 只需判定存储有字符信息的节点是否全部为叶子结点即可。若存储有某个字符信息的节点非叶子结点，即有子节点，那么它的 0/1 编码一定是它孩子节点 0/1 编码的前缀，违反了前缀特性。

2019 年统考 408 真题-数据结构部分

数据结构部分选择题：

1. 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（ ）。
 $x=0;$
 $\text{while } (n \geq (x+1) * (x+1))$
 $\quad x=x+1;$
 A. $O(\log n)$ B. $O(\sqrt{n})$ C. $O(n)$ D. $O(n^2)$
2. 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
3. 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
4. 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
5. 下图所示的 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（ ）。


```

graph LR
    1((1)) -- a=3 --> 2((2))
    1 -- b=4 --> 3((3))
    1 -- c=8 --> 3
    2 -- d=7 --> 4((4))
    2 -- e=6 --> 5((5))
    3 -- f=10 --> 5
    4 -- g=6 --> 6((6))
    5 -- h=9 --> 6
      
```

6. 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
7. 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

8. 现有长度为 11 且初始为空的散列表 HT，散列函数是 $H(\text{key}) = \text{key} \% 7$ ，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
9. 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（ ）。
 A. 9 B. 10 C. 12 D. 15
10. 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
11. 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

数据结构部分大题

12. (13 分) 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：

- (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。
13. (10 分) 请设计一个队列，要求满足：①初始时队列为空；②入队时，允许增加队列占用空间；③出队后，出队元素所占用的空间可重复使用，即整个队列所占用的空间只增不减；④入队操作和出队操作的时间复杂度始终保持为 $O(1)$ 。请回答下列问题：
- (1) 该队列应该选择链式存储结构，还是顺序存储结构？
 - (2) 画出队列的初始状态，并给出判断队空和队满的条件
 - (3) 画出第一个元素入队后的队列状态。
 - (4) 给出入队操作和出队操作的基本过程。

- | | | | | |
|------|------|------|------|-------|
| 1. B | 2. B | 3. C | 4. A | 5. C |
| 6. A | 7. D | 8. C | 9. B | 10. D |
11. B

12. 【答案要点】

(1) 算法的基本设计思想:

算法分 3 步完成。第 1 步, 采用两个指针交替前行, 找到单链表的中间结点; 第 2 步, 将单链表的后半段结点原地逆置; 第 3 步, 从单链表前后两段中依次各取一个结点, 按要求重排。

(2) 算法实现:

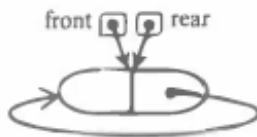
```
void change_list (NODE * h)
{
    NODE * p, * q, * r, * s;
    p=q=h;
    while (q->next != NULL)           //寻找中间结点
    {
        p=p->next;                      //p 走一步
        q=q->next;
        if (q->next != NULL) q=q->next; //q 走两步
    }
    q=p->next; //p 所指结点为中间结点, q 为后半段链表的首结点
    p->next=NULL;
    while (q != NULL) //将链表后半段逆置
    {
        r=q->next;
        q->next=p->next;
        p->next=q;
        q=r;
    }
    s=h->next; //s 指向前半段的第一个数据结点, 即插入点
    q=p->next; //q 指向后半段的第一个数据结点
    p->next=NULL;
    while (q != NULL) //将链表后半段的结点插入到指定位置
    {
        r=q->next; //r 指向后半段的下一个结点
        q->next=s->next; //将 q 所指结点插入到 s 所指结点之后
        s->next=q;
        s=q->next; //s 指向前半段的下一个插入点
        q=r;
    }
}
```

(3) 算法的时间复杂度:

参考答案的时间复杂度为 $O(n)$ 。

13. 【答案要点】

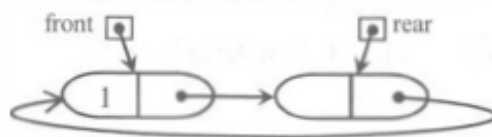
- (1) 采用链式存储结构（两段式单向循环链表），队头指针为 **front**，队尾指针为 **rear**。
- (2) 初始时，创建只有一个空闲结点的两段式单向循环链表，头指针 **front** 与尾指针 **rear** 均指向空闲结点。如下图所示。



队空的判定条件：**front==rear**。

队满的判定条件：**front==rear->next**。

- (3) 插入第一个元素后的队列状态：



- (4) 操作的基本过程：

入队操作

若 (**front==rear->next**) //队满

则在 **rear** 后面插入一个新的空闲结点；

入队元素保存到 **rear** 所指结点中；**rear=rear->next**；返回。

出队操作

若 (**front==rear**) //队空

则出队失败，返回；

取 **front** 所指结点中的元素 **e**；**front=front->next**；返回 **e**。

2018 年统考 408 真题-数据结构部分

数据结构部分选择题:

1. 若栈 S1 中保存整数, 栈 S2 中保存运算符, 函数 F()依次执行下述各步操作:

- (1) 从 S1 中依次弹出两个操作数 a 和 b;
- (2) 从 S2 中弹出一个运算符 op;
- (3) 执行相应的运算 $b \text{ op } a$;
- (4) 将运算结果压入 S1 中。

假定 S1 中的操作数依次是 5, 8, 3, 2 (2 在栈顶), S2 中的运算符依次是 *, -, + (+在栈顶)。调用 3 次 F()后, S1 栈顶保存的值是 ()。

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

2. 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 3 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

3. 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

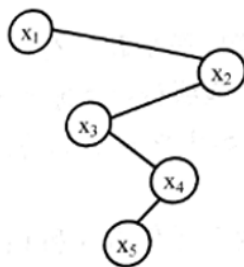
4. 设一棵非空完全二叉树 T 的所有叶结点均位于同一层, 且每个非叶结点都有 2 个子结点。若 T 有 k 个叶结点, 则 T 的结点总数是 ()。

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

5. 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 ()。

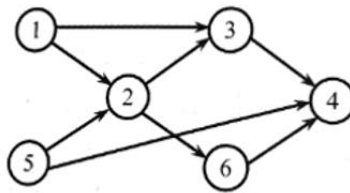
- A. 00, 1011, 01, 1010, 11, 100 B. 00, 100, 110, 000, 0010, 01
C. 10, 1011, 11, 0011, 00, 010 D. 0011, 10, 11, 0010, 01, 000

6. 已知二叉排序树如下图所示, 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. $x_4 < x_3 < x_5$

7. 下列选项中，不是如下有向图的拓扑序列的是（ ）。

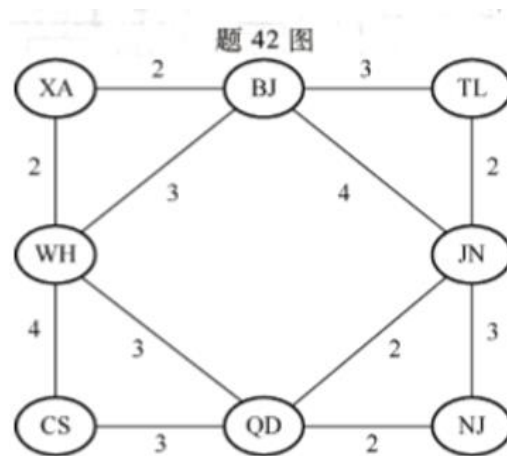


- A. 1,5,2,3,6,4 B. 5,1,2,6,3,4
C. 5,1,2,3,6,4 D. 5,2,1,6,3,4
8. 高度为 5 的 3 阶 B 树含有的关键字个数至少是（ ）。
- A. 15 B. 31 C. 62 D. 242
9. 现有长度为 7、初始为空的散列表 HT，散列函数 $H(k) = k \% 7$ ，用线性探测再散列法解决冲突。将关键字 22, 43, 15 依次插入到 HT 后，查找成功的平均查找长度是（ ）。
- A. 1.5 B. 1.6 C. 2 D. 3
10. 对初始数据序列（8, 3, 9, 11, 2, 1, 4, 7, 5, 10, 6）进行希尔排序。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
11. 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（ ）。
- 个数是
- 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

二、综合应用题

41. (13 分)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：
- (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。

42. (12 分)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，题 42 图中无向边上的权值表示两个城市间备选光缆的铺设费用。



请回答下列问题。

- (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) 题 42 图可采用图的哪一种存储结构？给出求解问题（1）所使用的算法名称

1. B 2. C 3. A 4. A 5. A
6. C 7. D 8. B 9. C 10. D
11. A

41. 【答案要点】

(1) 算法的基本设计思想:

设要查找的数组中未出现的最小正整数为 K 。 K 取值范围之内是 $[1, n+1]$ 。采用类似计数排序的思想分配一个数组 $B[n]$, 用来标记 A 中是否出现了 $1 \sim n$ 之间的正整数。

从左至右依次扫描数组元素 $A[i]$ 并标记数组 B 。若 $A[i]$ 是负数、零或是大于 n , 则忽略该值, 否则根据计数排序的思想将 $B[A[i]-1]$ 置为 1。

标记完毕, 遍历数组 B , 查找第一个值为 0 的元素, 其下标+1 即为目标元素 K ; 找不到 0 时, $K=n+1$ 。

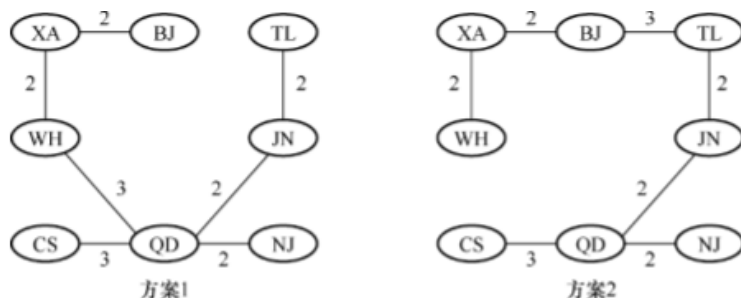
(2) 算法实现:

```
int findMissMin(int A[], int n)
{
    int i, *B;          // 标记数组
    B = (int *)malloc(sizeof(int)*n); // 分配空间
    memset(B, 0, sizeof(int)*n); // 赋初值为 0
    for(i=0; i<n; i++)
        if(A[i]>0 && A[i]<=n) // 若 A[i] 的值介于 1~n, 则标记数组 B
            B[A[i]-1] = 1;
    for(i=0; i<n; i++) // 扫描数组 B, 找到目标值
        if (B[i]==0) break;
    return i+1;        // 返回结果
}
```

(3) 算法的时间复杂度: 参考答案的时间复杂度为 $O(n)$ 。

42. 【答案要点】

(1) 为了求解最经济的方案, 可以把问题抽象为求无向带权图的最小生成树。可以采用手动 prim 算法或 kruskal 算法作图。注意本题最小生成树有两种构造, 如下图所示。



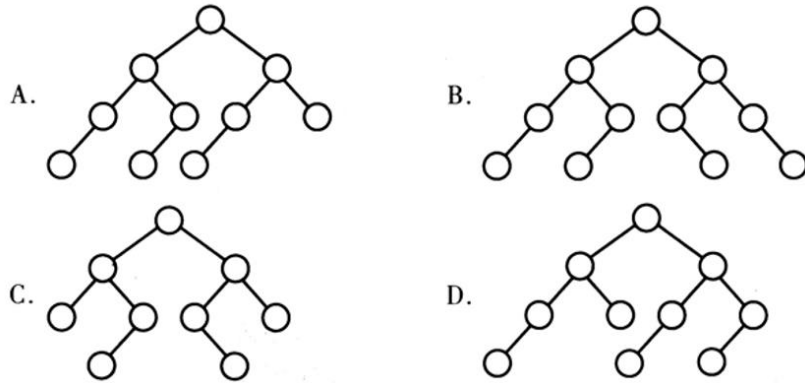
方案的总费用为 16。

(2) 存储题中的图可以采用邻接矩阵 (或邻接表)。构造最小生成树采用 Prim 算法 (或 kruskal 算法)。

8. 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

9. 下列二叉树中, 可能成为折半查找判定树(不含外部结点)的是 ()。



9. 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

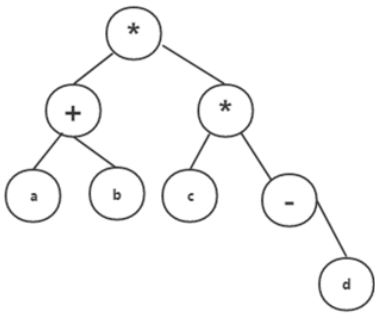
10. 在内部排序时, 若选择了归并排序而没有选择插入排序, 则可能的理由是 ()。

- 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

11. 下列排序方法中, 若将顺序存储更换为链式存储, 则算法时间效率降低的是 ()。

- I. 插入排序 II. 选择排序 III. 起泡排序
 - 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

41. (15 分)请设计一个算法，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算法的输入时：



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：

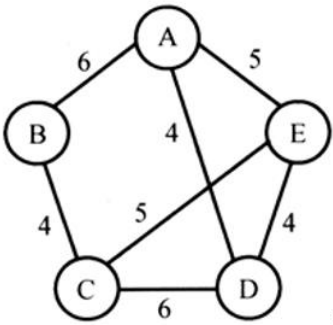
```
typedef struct node{
    char data[10];    //存储操作数或操作符
    struct node * left, *right;
} BTree;
```

要求：

- (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。

42. (8 分)使用 Prim(普里姆)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) 对下列图 G，从顶点 A 开始求 G 的 MST，依次给出按算法选出的边。
- (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？



1. B 2. C 3. A 4. B 5. B
6. D 7. B 8. B 9. A 10. B
11. D

41. 【答案要点】

(1) 算法的基本设计思想

表达式树的中序序列加上必要的括号即为等价的中缀表达式。可以基于二叉树的中序遍历策略得到所需的表达式。(3 分)

表达式树中分支结点所对应的子表达式的计算次序，由该分支结点所处的位置决定。为得到正确的中缀表达式，需要在生成遍历序列的同时，在适当位置增加必要的括号。显然，表达式的最外层(对应根结点)及操作数(对应叶结点)不需要添加括号。(2 分)

(2) 算法实现(10 分)

```
void BtreeToE(BTree * root){
    BtreeToExp(root, 1);    //根的高度为1
void BtreeToExp(BTree * root, int deep){
    if( root == NULL) return;
    else if(root->left == NULL && root->right == NULL)
        //若为叶结点
        printf("%s", root->data);    //输出操作数
    else
    {
        if(deep>1) printf("("); //若有子表达式则加1层括号
        BtreeToExp(root->left, deep+1);
        printf("%s", root->data);    //输出操作符
        BtreeToExp(root->right, deep+1);
        if(deep>1) printf(")"); //若有子表达式则加1层括号
    }
}
```

【评分说明】

①若考生设计的算法满足题目的功能要求，则(1)、(2)根据所实现算法的策略及输出结果给分，细则见下表。

分数	备注
15	采用中序遍历算法且正确，括号嵌套正确，层数适当。
14	采用中序遍历算法且正确，括号嵌套正确，但括号嵌套层数过多。例如，表达式最外层加上括号，或操作数加括号如(a)。
11	采用中序遍历算法，但括号嵌套层数不完全正确。例如，左右括号数量不匹配
9	采用中序遍历算法，但没有考虑括号。
≤7	其他

- ②若考生采用其他方法得到正确结果，可参照①的评分标准给分。
- ③如果程序中使用了求结点深度等辅助函数，但没有给出相应的实现过程，只要考生进行了必要的说明，可不扣分。
- ④若在算法的基本设计思想描述中因文字表达没有清晰反映出算法思路，但在算法实现中能够表达出算法思想且正确的，可参照①的标准给分。
- ⑤若算法的基本设计思想描述或算法实现中部分正确，可参照①中各种情况的相应给分标准酌情给分。
- ⑥参考答案中只给出了使用 C 语言的版本，使用 C++语言的答案参照以上评分标准。

42. 【答案要点】

(1)依次选出的边为：

(A, D), (D, E), (C, E), (B, C)(4 分)

【评分说明】每正确选对一条边且次序正确，给 1 分。若考生选择的边正确，但次序不完全正确，酌情给分。

(2) 图 G 的 MST 是唯一的。(2 分)

(3) 当带权连通图的任意一个环中所包含的边的权值均不相同，其 MST 是唯一的。(2 分)

【评分说明】

- ①若考生答案中给出的是其他充分条件，例如“带权连通图的所有边的权值均不相同”，同样给分。
- ②若考生给出的充分条件对图的顶点数和边数做了某些限制，例如，限制了图中顶点的个数(顶点个数少于 3 个)、限制了图的形状(图中没有环)等，则最高给 1 分。
- ③答案部分正确，酌情给分。

2016 年统考 408 真题-数据结构部分

1. 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	NUL
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

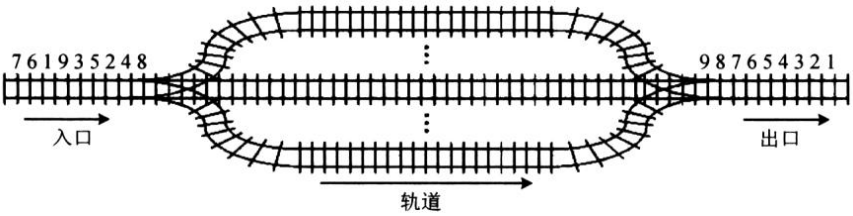
2. 已知一个带有表头结点的双向循环链表 L，结点结构为，

pr	da	ne
ev	ta	xt

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->next->prev=p->prev; p->prev->next=p->prev; free (p);
- B. p->next->prev=p->next; p->prey-> next=p->next; free (p);
- C. p->next->prev=p->next; p->prev->next=p->prev; free (p);
- D. p-> next-> prey=p->prey; p->prev->next=p->next; free (p);

3. 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

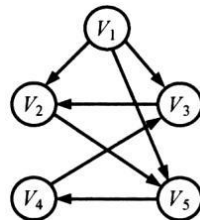
4. 有一个 100 阶的三对角矩阵 M ，其元素 $m_{ij}(1 \leq i \leq 100, 1 \leq j \leq 100)$ 按行优先次序压缩存入下标从 0 开始的一维数组 N 中。元素 $m_{30,30}$ 在 N 中的下标是 ()。

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

5. 若森林 F 有 15 条边、25 个结点，则 F 包含树的个数是 ()。

- A. 8 B. 9 C. 10 D. 11

6. 下列选项中，不是下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

7. 若将 n 个顶点 e 条弧的有向图采用邻接表存储，拓扑排序算法的时间复杂度是 ()。

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

8. 使用迪杰斯特拉(Dijkstra)算法求下图中从顶点 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

9. 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;
    else if(k-1<n 且 A[k-1]==x) 查找成功;
    else if(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 ()。

- 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

10. 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

11. 对 10 TB 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

42. (8 分)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 为正则后 k 树。请回答下列问题并给出推导过程。

(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?

43. (15 分)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:

- (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平均时间复杂度和空间复杂度。

1. D 2. D 3. C 4. B 5. C
 6. D 7. B 8. B 9. B 10. A
 11. D

42. 【答案要点】

(1)根据定义，正则 k 叉树中仅含有两类结点：叶结点(个数记为 n_0)和度为 k 的分支结点(个数记为 n_k)。树 T 中的结点总数 $n=n_0+n_k=n_0+m$ 。树中所含的边数 $e=n-1$ ，这些边均为 m 个度为 k 的结点发出的，即 $e=m \times k$ 。整理得 $n_0+m=m \times k+1$ ，故 $n_0=(k-1) \times m+1$ 。(3分)

(2)高度为 h 的正则 k 叉树 T 中，含最多结点的树形为：除第 h 层外，第 1 到第 $h-1$ 层的结点都是度为 k 的分支结点，而第 h 层均为叶结点，即树是“满”树。此时第 $j(1 \leq j \leq h)$ 层结点数为 k^{j-1} ，结点总数 M_1 为：

$$M_1 = \sum_{j=1}^h k^{j-1} = \frac{k^h - 1}{k - 1} \quad (3\text{分})$$

含最少结点的正则 k 叉树的树形为：第 1 层只有根结点，第 2 到第 $h-1$ 层仅含 1 个分支结点和 $k-1$ 个叶结点，第 h 层有 k 个叶结点。即除根外第 2 到第 h 层中每层的结点数均为 k ，故 T 中所含结点总数 M_2 为：

$$M_2 = 1 + (h-1) \times k \quad (2\text{分})$$

【评分说明】

- ①参考答案仅给出一种推导过程，若考生采用其他推导方法且正确，同样给分。
 ②若考生仅给出结果，但没有推导过程，则(1)、(2)的最高得分分别是 2 分和 3 分。若推导过程或答案不完全正确，酌情给分。

43.

(1) 算法的基本设计思想

由题意知，将最小的 $\lfloor \frac{n}{2} \rfloor$ 个元素放在 A1 中，其余的元素放在 A2 中，分组结果即

可满足题目要求。仿照快速排序的思想，基于枢轴将 n 个整数划分为两个子集。

根据划分后枢轴所处的位置 i 分别处理：

①若 $i = \lfloor \frac{n}{2} \rfloor$ ，则分组完成，算法结束；

②若 $i < \lfloor \frac{n}{2} \rfloor$ ，则枢轴及之前的所有元素均属于 A1，继续对 i 之后的元素进行划分；

③若 $i > \lfloor \frac{n}{2} \rfloor$ ，则枢轴及之后的所有元素均属于 A2，继续对 i 之前的元素进行划分；

基于该设计思想实现的算法，毋须对全部元素进行全排序，其平均时间复杂度是

$O(n)$ ，空间复杂度是 $O(1)$ 。

(2) 算法实现(9 分)

(3) 算法的平均时间复杂度和空间复杂度

时间复杂度是 $O(n)$

空间复杂度是 $O(1)$ 。

【评分说明】

①本题目只需将最大的一半元素与最小的一半元素分组，不需要对所有元素进行全部排序。参考答案基于快速排序思想，采用非递归的方式实现。若考生设计的算法满足题目的功能要求且正确，则(1)、(2)根据所实现算法的平均时间复杂度给分，细则见下表。

时间复杂度	分数	
$O(n)$	13	采用类似快排思想，没有对元素进行全排序
$O(n \log_2^n)$	11	
$O(n^2)$	9	
其他	7	时间复杂度高于 $O(n^2)$ 的算法

②若在算法的基本设计思想描述中因文字表达没有清晰反映出算法思路，但在算法实现中能够表达出算法思想且正确的，可参照①的标准给分。

③若算法的基本设计思想描述或算法实现中部分正确，可参照①中各种情况的相应给分标准酌情给分。

④参考答案中只给出了使用 C 语言的版本，使用 C++语言的答案视同使用 C 语言。

2015 年统考 408 真题-数据结构部分

1. 已知程序如下：

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

void main(){
    cout<<S(1);
}
```

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（ ）。

- A. main()→S(1)→S(0)
- B. S(0)→S(1)→main()
- C. main()→S(0)→S(1)
- D. S(1)→S(0)→main()

2. 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

3. 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

4. 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

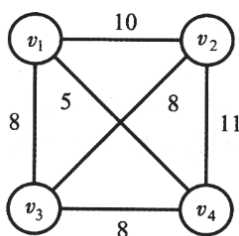
5. 设有向图 $G=(V, E)$ ，顶点集 $V=\{v_0, v_1, v_2, v_3\}$ ，边集 $E: \{<v_0, v_1>, <v_0, v_2>, <v_0, v_3>, <v_1, v_3>\}$ 。

若从顶点 v_0 开始对图进行深度优先遍历，则可能得到的不同遍历序列个数是（ ）。

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

6. 求下面带权图的最小(代价)生成树时，可能是克鲁斯卡尔(Kruskal)算法第 2 次选中但不 是普里姆(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)

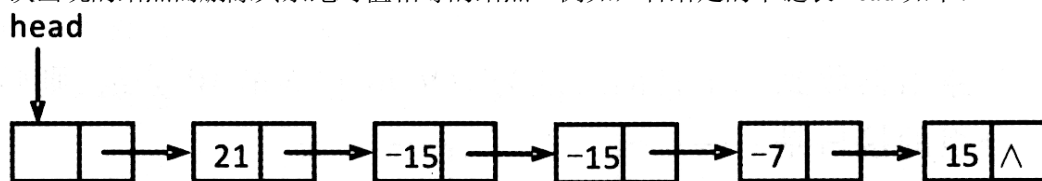


7. 下列选项中，不能构成折半查找中关键字比较序列的是（ ）。
- A. 500, 200, 450, 180 B. 500, 450, 200, 180
- C. 180, 500, 200, 450 D. 180, 200, 500, 450
8. 已知字符串 s 为“abaabaabacacaabaabcc”，模式串 t 为“abaabc”。采用 KMP 算法进行匹配，第一次出现“失配”($s[i] \neq t[j]$)时， $i=j=5$ ，则下次开始匹配时 i 和 j 的值分别是（ ）。
- A. $i=1, j=0$ B. $i=5, j=0$
- C. $i=5, j=2$ D. $i=6, j=2$
9. 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. 起泡排序
- C. 基数排序 D. 快速排序
10. 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
11. 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

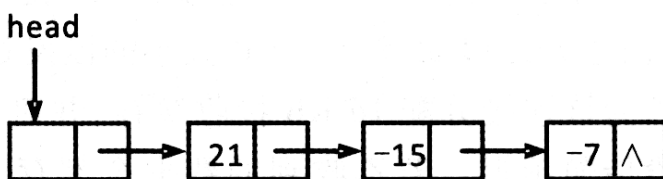
41.(15 分)用单链表保存 m 个整数，结点的结构为：

data	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：



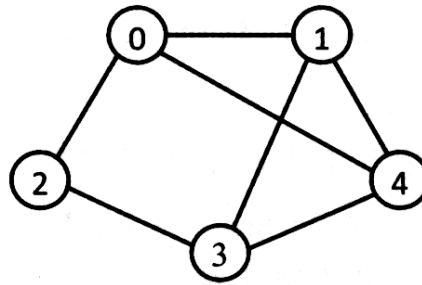
则删除结点后的 $head$ 为：



- (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。

42. (8 分) 已知含有 5 个顶点的图 G 如下图所示。请回答下列问题。

(1) 写出图 G 的邻接矩阵 A (行、列下标均从 0 开始)。



(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?

- | | | | | |
|-------|------|------|------|-------|
| 1. A | 2. B | 3. D | 4. D | 5. D |
| 6. C | 7. A | 8. C | 9. C | 10. C |
| 11. A | | | | |

41. (1)算法的基本设计思想

算法的核心思想是用空间换时间。使用辅助数组记录链表中已出现的数值，从而只需对链表进行一趟扫描。

因为 $|data| \leq n$ ，故辅助数组 q 的大小为 $n+1$ ，各元素的初值均为 0。依次扫描链表中的各结点，同时检查 $q[|data|]$

的值，如果为 0，则保留该结点，并令 $q[|data|]=1$ ；否则，将该结点从链表中删除。

(2) 使用 C 语言描述的单链表结点的数据类型定义

```

1  #include <stdio.h>
2  #include<limits.h>
3  #include <stdlib.h>
4
5  //灰灰考研
6  typedef struct ListNode
7  {
8      int data;
9      struct Node * pNext;
10 }NODE, *PNODE;
```

(3)算法实现

【略】

【(1)、(2)、(3)的评分说明】

①若考生设计的算法满足题目的功能要求且正确，则(1)、(3)根据所实现算法的时间复杂度给分，细则见下表：

时间复杂度	分数	说明
$O(m)$	11	对链表进行一趟扫描，且采用时间复杂度为 $O(1)$ 的方法判断 $ data $ 是否是首次出现。
$O(m)$	8	对链表进行一趟扫描，且采用时间复杂度 $>O(1)$ 的方法判断 $ data $ 是否是首次出现。
$>O(m)$	5	对链表进行多趟扫描，例如采用顺序查找算法，对每个结点在 当前链表中进行查找，删除重复出现的结点。

②若在算法的基本设计思想描述中因文字表达没有清晰反映出算法思路，但在算法实现中能够表达出算法思想且正确的，可参照①的标准给分。

③若算法的基本设计思想描述或算法实现中部分正确，可参照①中各种情况的相应给分标准酌情给分。

④若考生给出的单链表结点的数据类型定义与题目中所给的结点形式不完全相同，酌情给分。

⑤参考答案中只给出了使用 C 语言的版本，使用 C++语言的答案视同使用 C 语言。

(4)参考答案所给算法的时间复杂度为 $O(m)$ ，空间复杂度为 $O(n)$ 。

42.

(1) 图 G 的邻接矩阵 A 如下：

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

(2) A^2 如下：

$$A^2 = \begin{bmatrix} 3 & 1 & 0 & 3 & 1 \\ 1 & 3 & 2 & 1 & 2 \\ 0 & 2 & 2 & 0 & 2 \\ 3 & 1 & 0 & 3 & 1 \\ 1 & 2 & 2 & 1 & 3 \end{bmatrix}$$

0行3列的元素值3表示从顶点0到顶点3之间长度为2的路径共有3条。

(3) $B^m(2 \leq m \leq n)$ 中位于 i 行 j 列 ($0 \leq i, j \leq n-1$) 的非零元素的含义是：图中从顶点 i 到顶点 j 长度为 m 的路径条数。

【评分说明】

①若考生给出的邻接矩阵 A 中，结点与行、列的对应次序与本参考答案不完全一致，只要正确，同样给分。问题(2)中，考生所给的答案中顶点编号要与其所给的邻接矩阵相对应。

②若考生给出的矩阵 A 及 A^2 部分正确，酌情给分。

③若考生分别说明矩阵 B^2 、 B^3 、...、 B^m 中非零元素的含义，同样给分。

④若考生给出的 $B^m(2 \leq m \leq n)$ 中非零元素的含义部分正确，酌情给分。

2014 年统考 408 真题-数据结构部分

1. 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. $O(n^2)$

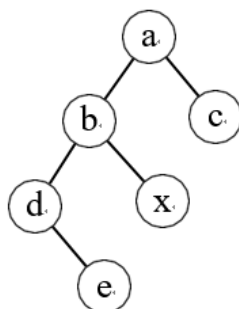
2. 假设栈初始为空, 将中缀表达式 $a/b+(c*d-e*f)/g$ 转换为等价的后缀表达式的过程中, 当扫描到 f 时, 栈中的元素依次是 ()。

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

3. 循环队列放在一维数组 $A[0 \cdots M-1]$ 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: $end2 == (end1+1) \bmod (M-1)$

4. 若对如下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

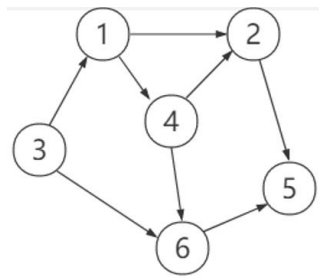
5. 将森林 F 转换为对应的二叉树 T , F 中叶结点的个数等于 ()。

- 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

6. 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

7. 对如下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
8. 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
9. 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
10. 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

11. 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

41. (13 分) 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，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：

- 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。

- | | | | | |
|-------|------|------|------|-------|
| 1. C | 2. B | 3. A | 4. D | 5. C |
| 6. D | 7. D | 8. D | 9. D | 10. B |
| 11. C | | | | |

41.

【评分说明】

- ①若考生给出能够满足题目要求的其他算法，且正确，可同样给分。
- ②考生答案无论使用 C 或 C++ 语言，只要正确同样给分。
- ③若对算法的基本设计思想和主要数据结构描述不十分准确，但在算法实现中能够清晰反映出算法思想且正确，参照①的标准给分。
- ④若考生给出的二叉树结点的数据类型定义和算法实现中，使用的是除整型之外的其他数值，可视同使用整型类型。
- ⑤若考生给出的答案中算法主要设计思想或算法中部分正确，可酌情给分。

在层次遍历的算法中，读者要理解 lastNode 和 newlastNode 的区别，lastNode 指的是当前遍历层的最后一个结点，而 newlastNode 指的是下一层的最后一个结点，是动态变化的，直到遍历到本层的最后一个结点，才能确认下层真正的最后一个结点是哪个结点，而函数中入队操作并没有判断队满，若考试时用到，读者最好加上队满条件，这里队列的队满条件为 $end1 == (end2 + 1) \% M$ ，采用的是 2014 年真题选择题中第三题的队列形式。同时，考生也可以尝试使用记录每层的第一个结点来进行层次遍历的算法，这里不再给出代码，请考生自行练习。

(1) 算法的基本设计思想

基于层次遍历的算法思想是使用队列进行层次遍历，并记录当前的层数，当遍历到叶子结点时，累计 wpl；

当遍历到非叶子结点时对该结点的把该结点的子树加入队列；当某结点为该层的最后一个结点时，层数自增 1； 队列空时遍历结束，返回 wpl。

(2) 二叉树结点的数据类型定义如下：

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  //灰灰考研
5  //二叉树结点定义
6  typedef struct BiTNode
7  {
8      int weight;
9      struct BiTNode *lchild,*rchild;
10 }BiTNode,*BiTree;
```

(3) 算法代码如下：

略

2013 年统考 408 真题-数据结构部分

1. 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. $O(m*n)$ C. $O(\min(m,n))$ D. $O(\max(m,n))$

2. 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 为 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

3. 若将关键字 $1, 2, 3, 4, 5, 6, 7$ 依次插入到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

4. 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

5. 若 X 是后序线索二叉树中的叶结点， X 存在左兄弟结点 Y ，则 X 右线索指的是（ ）。

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

6. 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（ ）。

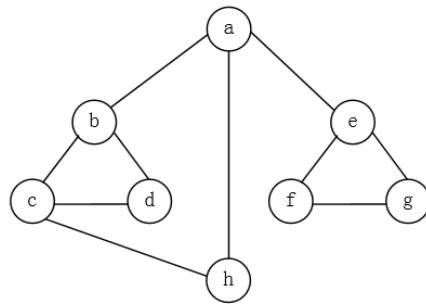
- I. 若 v 是 T_1 的叶结点，则 T_1 与 T_3 不同
II. 若 v 是 T_1 的叶结点，则 T_1 与 T_3 相同
III. 若 v 不是 T_1 的叶结点，则 T_1 与 T_3 不同
IV. 若 v 不是 T_1 的叶结点，则 T_1 与 T_3 相同
A. 仅 I、III B. 仅 I、IV C. 仅 II、III D. 仅 II、IV

7. 设图的邻接矩阵 A 如下所示。各顶点的度依次是（ ）。

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

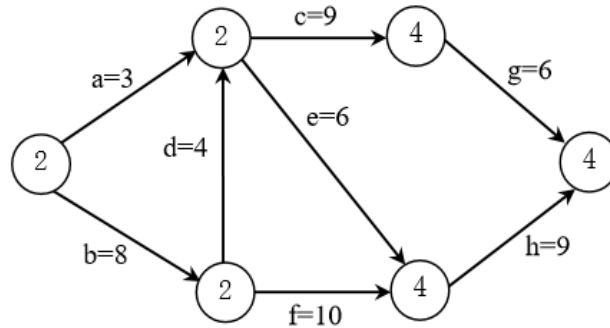
- A. 1, 2, 1, 2
B. 2, 2, 1, 1
C. 3, 4, 2, 3
D. 4, 4, 2, 2

8. 若对如下无向图进行遍历，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

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



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

10. 在一株高度为 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

11. 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

41. (13 分) 已知一个整数序列 $A(a_0, a_1, a_2, \dots, a_n)$, 其中 $0 \leq a_i \leq n (0 \leq i \leq n)$ 。若存在 $a_{p_1} = a_{p_2} = \dots = a_{p_m} = x$ 且 $m > n/2 (0 \leq p_k \leq n, 1 \leq k \leq m)$, 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:

- (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。

42. (10 分) 设包含 4 个数据元素的集合 $S = \{\text{"do"}, \text{"for"}, \text{"repeat"}, \text{"while"}\}$, 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:

- (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?

- | | | | | |
|-------|------|------|------|-------|
| 1. D | 2. C | 3. D | 4. B | 5. A |
| 6. C | 7. C | 8. D | 9. C | 10. A |
| 11. C | | | | |

41. 【答案要点】

(1) 给出算法的基本设计思想：(4 分) 算法的策略是从前向后扫描数组元素，标记出一个可能成为主元素的元素 Num。然后重新计数，确认 Num 是否是主元素。算法可分为以下两步：

①选取候选的主元素：依次扫描所给数组中的每个整数，将第一个遇到的整数 Num 保存到 c 中，记录 Num 的出现次数为 1；若遇到的下一个整数仍等于 Num，则计数加 1，否则计数减 1；当计数减到 0 时，将遇到的下一个整数保存到 c 中，计数重新记为 1，开始新一轮计数，即从当前位置开始重复上述过程，直到扫描完全部数组元素。

②判断 c 中元素是否是真正的主元素：再次扫描该数组，统计 c 中元素出现的次数，若大于 $n/2$ ，则为主元素；否则，序列中不存在主元素。

(2) 算法实现：(7 分)

```
int Majority(int A[], int n) {
    int i, c, count=1;    //c 用来保存候选主元素，count 用来计数
    c=A[0]; //设置 A[0] 为候选主元素
    for(i=1; i<n; i++)    //查找候选主元素
        if(A[i]==c)
            count++; //对 A 中的候选主元素计数
        else
            if(count>0) //处理不是候选主元素的情况
                count--;
            else //更换候选主元素，重新计数
            {
                c=A[i];
            }
}
```

```

if(count>0)
    count=1;
for(i=count=0;i<n;i++) //统计候选主元素的实际出现次数
    if(A[i]==c)
        count++;
if(count>n/2)
    return c; //确认候选主元素
else
    return -1; //不存在主元素
}

```

【(1)、(2) 的评分说明】

①若考生设计的算法满足题目的功能要求且正确，则 (1)、(2) 根据所实现算法的效率给分，细则见下表：

时间复杂度	空间复杂度	(1) 得分	(2) 得分	说明
$O(n)$	$O(1)$	4	7	
$O(n)$	$O(n)$	4	6	如采用计数排序思想， 见表后 Majority1 程序
$O(n\log_2 n)$	其他	3	6	如采用其他排序的思想
$\geq O(n^2)$	其他	3	5	其他方法

```

int Majority1(int A[],int n)//采用计数排序思想，时间： $O(n)$ ，空间： $O(n)$ 
{
    int k,*p,max;
    p=(int *)malloc(sizeof(int)*n);    //申请辅助计数数组
    for(k=0;k<n;k++)p[k]=0;    //计数数组清 0max=0;
    for(k=0;k<n;k++) {
        p[A[k]]++; //计数器+1
        if(p[A[k]]>p[max])
            max=A[k]; //记录出现次数最多的元素
    }
}

```



```

if (p[max]>n/2) return max;
else return -1;
}

```

②若在算法的基本设计思想描述中因文字表达没有非常清晰反映出算法思路，但在算法实现中能够清晰看出算法思想且正确的，可参照①的标准给分。

③若算法的基本设计思想描述或算法实现中部分正确，可参照①中各种情况的相应给分标准酌情给分。

④参考答案中只给出了使用 C 语言的版本，使用 C++ 或 Java 语言的答案视同使用 C 语言。

(3) 说明算法复杂性：(2 分)

参考答案中实现的程序的时间复杂度为 $O(n)$ ，空间复杂度为 $O(1)$ 。

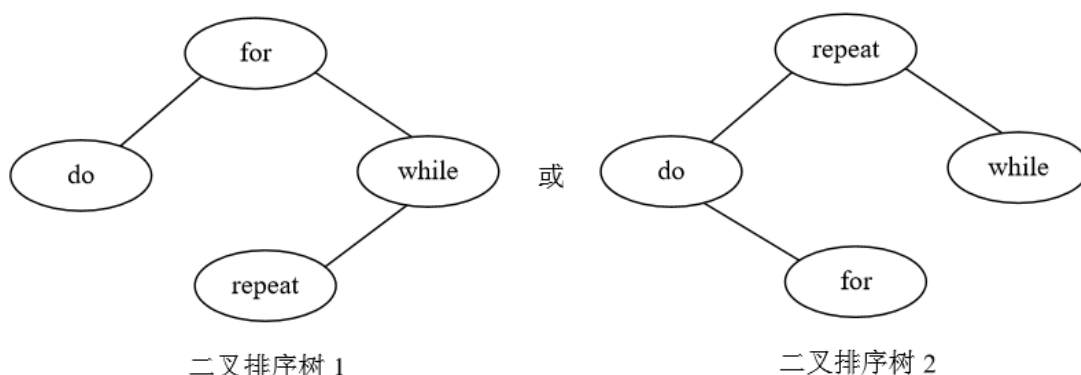
【评分说明】若考生所估计的时间复杂度与空间复杂度与考生所实现的算法一致，可各给 1 分。

42. 【答案要点】

(1) 采用顺序存储结构，数据元素按其查找概率降序排列。(2 分) 采用顺序查找方法。(1 分) 查找成功时的平均查找长度 $= 0.35 \times 1 + 0.35 \times 2 + 0.15 \times 3 + 0.15 \times 4 = 2.1$ 。(2 分)

(2) 【答案一】采用链式存储结构，数据元素按其查找概率降序排列，构成单链表。(2 分) 采用顺序查找方法。(1 分) 查找成功时的平均查找长度 $= 0.35 \times 1 + 0.35 \times 2 + 0.15 \times 3 + 0.15 \times 4 = 2.1$ 。(2 分)

【答案二】采用二叉链表存储结构，构造二叉排序树，元素存储方式见下图。(2 分)



查找成功时的平均查找长度 $= 0.15 \times 1 + 0.35 \times 2 + 0.35 \times 2 + 0.15 \times 3 = 2.0$ 。(2 分)

2012 年统考 408 真题-数据结构部分

1. 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. $O(n^2)$

2. 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

3. 若一棵二叉树的前序遍历序列为 a, e, b, d, c ，后序遍历序列为 b, c, d, e, a 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

4. 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. 10 B. 20 C. 32 D. 33

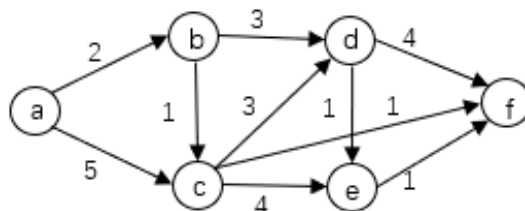
5. 对有 n 个结点、 e 条边且使用邻接表存储的有向图进行广度优先遍历，其算法时间复杂度是 ()。

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

6. 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

7. 对如下有向带权图，若采用迪杰斯特拉 (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 ()。

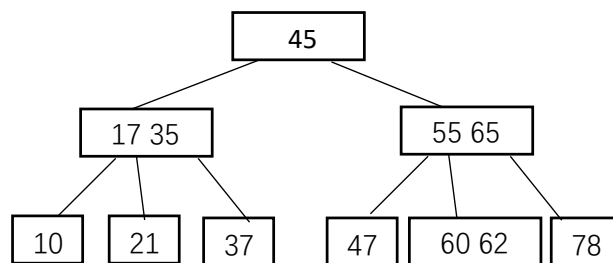


- A. d, e, f B. e, d, f C. f, d, e D. f, e, d

8. 下列关于最小生成树的叙述中, 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. 使用普里姆 (Prim) 算法从不同顶点开始得到的最小生成树一定相同
 - IV. 使用普里姆算法和克鲁斯卡尔 (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. 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, 其最右叶结点中的关键字是 ()。



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

10. 在内部排序过程中, 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最终位置的方法是 ()。

- 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

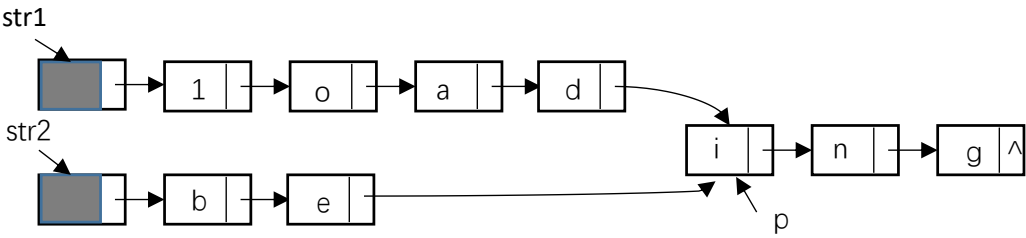
11. 对一待排序序列分别进行折半插入排序和直接插入排序, 两者之间可能的不同之处是 ()。

- A. 排序的总趟数 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

41. 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达到最小。请问答下列问题。

- 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。

42. 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：

- 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。

- | | | | | |
|-------|------|------|------|-------|
| 1. B | 2. A | 3. A | 4. B | 5. C |
| 6. C | 7. C | 8. A | 9. D | 10. A |
| 11. D | | | | |

41. 解答:

本题同时对多个知识点进行了综合考查。对有序表进行两两合并考查了归并排序中的 Merge() 函数; 对合并过程的设计考查了哈夫曼树和最佳归并树。外部排序属于大纲新增考点。

1) 对于长度分别为 m, n 的两个有序表的合并, 最坏情况下是一直比较到两个表尾元素, 比较次数为 $m+n-1$ 次。故, 最坏情况的比较次数依赖于表长, 为了缩短总的比较次数, 根据哈夫曼树(最佳归并树)思想的启发, 可采用如图所示的合并顺序。

根据上图中的哈夫曼树, 6 个序列的合并过程为:

第 1 次合并: 表 A 与表 B 合并, 生成含有 45 个元素的表 AB;

第 2 次合并: 表 AB 与表 C 合并, 生成含有 85 个元素的表 ABC;

第 3 次合并: 表 D 与表 E 合并, 生成含有 110 个元素的表 DE;

第 4 次合并: 表 ABC 与表 DE 合并, 生成含有 195 个元素的表 ABCDE; 第 5 次合并: 表 ABCDE 与表 F 合并, 生成含有 395 个元素的最终表。

由上述分析可知, 最坏情况下的比较次数为: 第 1 次合并, 最多比较次数 $=10+35-1=44$; 第 2 次合并, 最多比较次数 $=45+40-1=84$; 第 3 次合并, 最多比较次数 $=50+60-1=109$; 第 4 次合并, 最多比较次数 $=85+110-1=194$; 第 5 次合并, 最多比较次数 $=195+200-1=394$ 。故比较总次数最多为: $44+84+109+194+394=825$ 。

2) 各表的合并策略是: 在对多个有序表进行两两合并时, 若表长不同, 则最坏情况下总的比较次数依赖于表的合并次序。可以借用哈夫曼树的构造思想, 依次选择最短的两个表进行合并, 可以获得最坏情况下最佳的合并效率。

【1) 2) 评分说明】

①对于用类似哈夫曼树(或最佳归并树)思想进行合并, 过程描述正确, 给 5 分。按其他策略进行合并, 过程描述正确, 给 3 分。

②正确算出与合并过程一致的总比较次数, 给 2 分。若计算过程正确, 但结果错误, 可给 1 分。

③考生只要说明采用的是类似哈夫曼树(或最佳归并树)的构造方法作为合并策略, 即可给 3 分。如果采用其他策略, 只要能够完成合并, 给 2 分。

42. 解答:

1) 顺序遍历两个链表到尾结点时, 并不能保证两个链表同时到达尾结点。这是因为两个链表的长度不同。假设一个链表比另一个链表长 k 个结点, 我们先在长链表上遍历 k 个结点, 之后同步遍历两个链表, 这样就能够保证它们同时到达最后一个结点。由于两个链表从第一个公共结点到链表的尾结点都是重合的, 所以它们肯定同时到达第一个公共结点。算法的基本设计思想:

①分别求出 str1 和 str2 所指的两个链表的长度 m 和 n ;

②将两个链表以表尾对齐：令指针 p、q 分别指向 str1 和 str2 的头结点，若 $m \geq n$ ，则使 p 指向链表中的第 $m-n+1$ 个结点；若 $m < n$ ，则使 q 指向链表中的第 $n-m+1$ 个结点，即使指针 p 和 q 所指的结点到表尾的长度相等。

③反复将指针 p 和 q 同步向后移动，并判断它们是否指向同一结点。若 p 和 q 指向同一结点，则该点即为所求的共同后缀的起始位置。

2) 算法的 C 语言代码描述：

```
LinkNode* Find_1st_Common(LinkListstr1, LinkListstr2) {  
  
    int len1=Length(str1), len2=Length(str2);  
  
    LinkNode *p, *q;  
  
    for(p=str1; len1>len2; len1--) //使 p 指向的链表与 q 指向的链表等长  
        p=p->next;  
  
    for(q=str2; len1<len2; len2--) //使 q 指向的链表与 p 指向的链表等长  
        q=q->next;  
  
    while(p->next!=NULL && p->next!=q->next) { //查找共同后缀起始点  
        p=p->next; //两个指针同步向后移动  
        q=q->next;  
    }  
  
    return p->next; //返回共同后缀的起始点  
}
```

【1) 2) 的评分说明】

①若考生所给算法实现正确，且时间复杂度为 $O(m+n)$ ，可给 12 分；若算法正确，但时间复杂度超过 $O(m+n)$ ，则最高可给 9 分。

②若在算法的基本设计思想描述中因文字表达没有非常清晰反映出算法思路，但在算法实现中能够清晰看出算法思想且正确的，可参照①的标准给分。

③若算法的基本设计思想描述或算法实现中部分正确，可参照①中各种情况的相应给分标准酌情给分。

④参考答案中只给出了使用 C 语言的版本，使用 C++/JAVA 语言的答案视同时用 C 语言。

3) 时间复杂度为： $O(len1+len2)$ 或 $O(\max(len1, len2))$ ，其中 len1、len2 分别为两个链表的长度。

【3) 的评分说明】若考生所估计的时间复杂度与考生所实现的算法一致，可给 1 分。

2011 年统考 408 真题-数据结构部分

1. 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.  $O(n^2)$
2. 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,直到所有的元素都出栈,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
3. 已知循环队列存储在一维数组  $A[0..n-1]$  中,且队列非空时  $front$  和  $rear$  分别指向队头和队尾元素。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.  $n-1, n-1$
4. 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
5. 若一棵二叉树的前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
6. 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的个数是 ( )。
- A. 115      B. 116      C. 1895      D. 1896
7. 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
8. 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
9. 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

10. 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

11. 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

41. (8 分) 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：

- (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。

42. (15 分) 一个长度为 L ( $L \geq 1$ ) 的升序序列 S，处在第  $L/2$  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：

- (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。



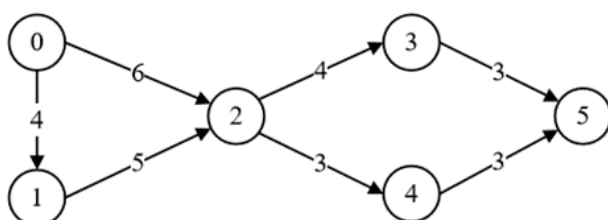
- |       |      |      |      |       |
|-------|------|------|------|-------|
| 1. A  | 2. B | 3. B | 4. C | 5. C  |
| 6. D  | 7. A | 8. C | 9. D | 10. A |
| 11. B |      |      |      |       |

41. 解答:

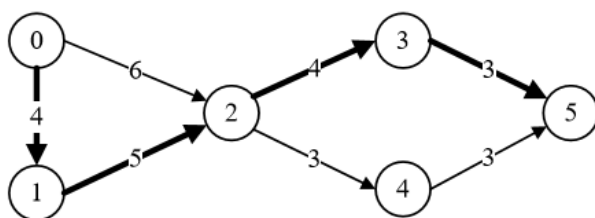
(1) 图 G 的邻接矩阵 A 如下所示:

$$A = \begin{pmatrix} 0 & 4 & 6 & \infty & \infty & \infty \\ \infty & 0 & 5 & \infty & \infty & \infty \\ \infty & \infty & 0 & 4 & 3 & \infty \\ \infty & \infty & \infty & 0 & \infty & 3 \\ \infty & \infty & \infty & \infty & 0 & 3 \\ \infty & \infty & \infty & \infty & \infty & 0 \end{pmatrix}$$

(2) 有向带权图 G 如下图所示:



(3) 关键路径为 01235 (如下图所示粗线表示), 长度为  $4+5+4+3=16$



42. 解答:

(1) 算法的基本设计思想如下。

分别求出序列 A 和 B 的中位数, 设为 a 和 b, 求序列 A 和 B 的中位数过程如下:

1) 若  $a=b$ , 则 a 或 b 即为所求中位数, 算法结束。

2) 若  $a < b$ , 则舍弃序列 A 中较小的一半, 同时舍弃序列 B 中较大的一半, 要求舍弃的长度相等; 在保留的两个升序序列中, 重复过程 1)、2)、3), 直到两个序列中只含一个元素时为止, 较小者即为所求的中位数。

(2) 算法的实现如下:

```
Int M_Search(intA[], intB[], intn) {
 ints1=0, d1=n-1, m1, s2=1, d2=n-1, m2;
```

//分别表示序列 A 和 B 的首位数、末位数和中位数

```
while(s1!=d1||s2!=
 d2) {m1=(s1+d1)/
 2;m2=(s2+d2)/2;
 if(A[m1]==B[m2])
 returnA[m1]; //满足条件 1)
 if(A[m1]<B[m2]) { //满足条
 件 2)
 if((s1+d1)%2==0) { //若元素个数为奇数
 s1=m1; //舍弃 A 中间点以前的部分且保留中间点
 d2=m2; //舍弃 B 中间点以后的部分且保留中间点
 }
 else { //元素个数为偶数
 s1=m1+1; //舍弃 A 中间点及中间点以前部分
 d2=m2; //舍弃 B 中间点以后部分且保留中间点
 }
 }
 else { //满足条件 3)
 if((s1+d1)%2==0) { //若元素个数为奇数
 d1=m1; //舍弃 A 中间点以后的部分且保留中间点
 s2=m2; //舍弃 B 中间点以前的部分且保留中间点
 }
 else { //元素个数为偶数
 d1=m1+1; //舍弃 A 中间点以后部分且保留中间点
 s2=m2; //舍弃 B 中间点及中间点以前部分
 }
 }
}
returnA[s1]<B[s2]?A[s1]:B[s2];
}
```

## 2010 年统考 408 真题-数据结构部分

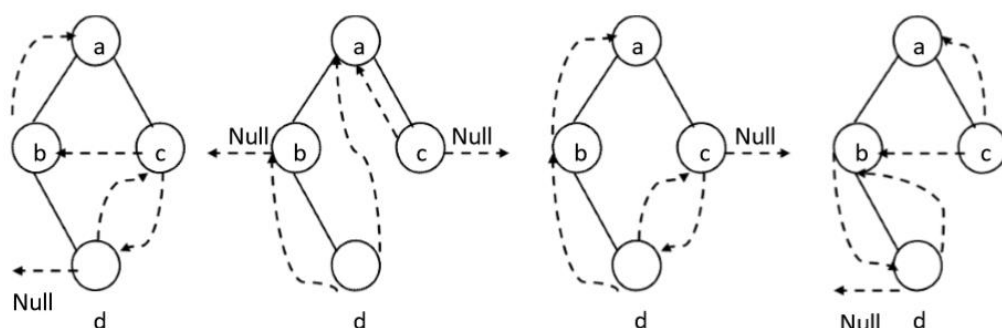
1. 若元素 a、b、c、d、e、f 依次进栈，允许进栈、退栈操作交替进行，但不允许连续三次进行退栈操作，则不可能得到的出栈序列是（ ）。

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

2. 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（ ）。

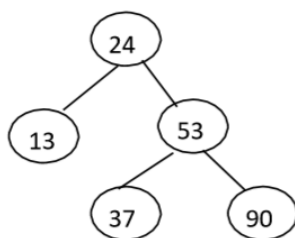
- A. bacde      B. dbace      C. dbcae      D. ecbad

3. 下列线索二叉树中（用虚线表示线索），符合后序线索树定义的是（ ）。



- A.      B.      C.      D.

4. 在下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（ ）。



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

5. 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（ ）。

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

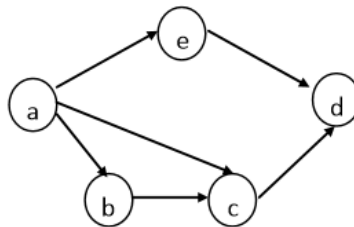
6. 对  $n$  ( $n \geq 2$ ) 个权值均不相同的字符构造哈夫曼树。下列关于该哈夫曼树的叙述中，错误的是（ ）。

- 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。

7. 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

8. 对下图进行拓扑排序, 可以得到不同的拓扑序列的个数是 ( )。



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

9. 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

10. 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 ( )。

- 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处理顺序无关。

11. 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

则采用的排序方法可能是 ( )。

- A. 起泡排序              B. 希尔排序              C. 归并排序              D. 基数排序

41. (10 分) 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:  $H(\text{key})=(\text{key} \times 3) \text{MOD} 7$ , 处理冲突采用线性探测再散列法, 要求装填 (载) 因子为 0.7。

(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。

42. (13 分) 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:

- (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。

1. D                  2. C                  3. D                  4. C                  5. B  
6. A                  7. C                  8. B                  9. B                  10. A  
11. A

41. 解答：

(1) 由装载因子 0.7，数据总数为 7，得一维数组大小为  $7/0.7=10$ ，数组下标为  $0\sim 9$ 。所构造的散列函数值如下所示：

|        |   |   |    |    |    |   |    |
|--------|---|---|----|----|----|---|----|
| key    | 7 | 8 | 30 | 11 | 18 | 9 | 14 |
| H(key) | 0 | 3 | 6  | 5  | 5  | 6 | 0  |

采用线性探测再散列法处理冲突，所构造的散列表为：

|     |   |    |   |   |   |    |    |    |   |   |
|-----|---|----|---|---|---|----|----|----|---|---|
| 地址  | 0 | 1  | 2 | 3 | 4 | 5  | 6  | 7  | 8 | 9 |
| 关键字 | 7 | 14 |   | 8 |   | 11 | 30 | 18 | 9 |   |

(2) 查找成功时，是根据每个元素查找次数来计算平均长度，在等概率的情况下，各关键字的查找次数为：

|     |   |   |    |    |    |   |    |
|-----|---|---|----|----|----|---|----|
| key | 7 | 8 | 30 | 11 | 18 | 9 | 14 |
| 次数  | 1 | 1 | 1  | 1  | 3  | 3 | 2  |

故， $ASL_{成功} = \text{查找次数} / \text{元素个数} = (1+2+1+1+1+3+3)/7 = 12/7$

这里要特别防止惯性思维。查找失败时，是根据查找失败位置计算平均次数，根据散列函数 MOD7，初始只可能在  $0\sim 6$  的位置。等概率情况下，查找  $0\sim 6$  位置查找失败的查找次数为：

|        |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|
| H(key) | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 次数     | 3 | 2 | 1 | 2 | 1 | 5 | 4 |

故， $ASL_{不成功} = \text{查找次数} / \text{散列后的地址个数} = (3+2+1+2+1+5+4)/7 = 18/7$

42. 解答：

(1) 算法的基本设计思想：

可以将这个问题看做是把数组 ab 转换成数组 ba (a 代表数组的前 p 个元素，b 代表数组中余下  $n-p$  个元素)，先将 a 逆置得到  $a^{-1}b$ ，再将 b 逆置得到  $a^{-1}b^{-1}$ ，最后将整个  $a^{-1}b^{-1}$  逆置得到  $(a^{-1}b^{-1})^{-1} = ba$ 。设 Reverse 函数执行将数组元素逆置的操作，对 abcdefgh 向左循环移动 3 ( $p=3$ ) 个位置的过程如下：

Reverse(0, p-1) 得到 cbadefgh；

Reverse(p, n-1) 得到 cbahgfed；

Reverse(0, n-1) 得到 defghabc；

注：Reverse 中，两个参数分别表示数组中待转换元素的始末位置。

(2) 使用 c 语言 描述算法如下：

```
void Reverse(int R[], int from, int to) {
 int i, temp;
 for(i=0; i<(to-from+1)/2; i++) {
 temp=R[from+i];
 R[from+i]=R[to-i];
 R[to-i]=temp;}
} // Reverse
```

```
void Converse(int R[], int n, int p) {
 Reverse(R, 0, p-1);
 Reverse(R, p, n-1);
 Reverse(R, 0, n-1);
}
```

(3) 上述算法的时间复杂度为  $T(n)$ , 空间复杂度为  $O(1)$ 。

另解, 借助辅助数组来实现:  $(n/2)$ , 故所设计算法思想: 创建大小为  $p$  的辅助数组  $S$ , 将  $R$  中前  $p$  个整数依次暂存在  $S$  中, 同时将  $R$  中后  $n-p$  个整数左移, 然后将  $S$  中暂存的  $p$  个数依次放回到  $R$  中的后续单元。时间复杂度为  $T(n)$ , 空间复杂度为  $O(p)$ 。

## 2009 年统考 408 真题-数据结构部分

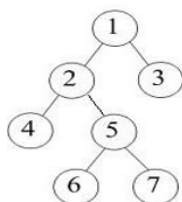
1. 为解决计算机主机与打印机之间速度不匹配问题，通常设置一个打印数据缓冲区的主机将要输出的数据依次写入该缓冲区，而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是（ ）。

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

2. 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  $S$  中。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（ ）。

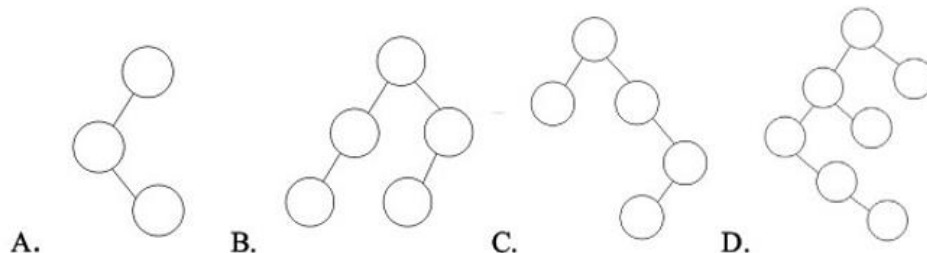
- A. 1                      B. 2                      C. 3                      D. 4

3. 给定二叉树如下图所示。设  $N$  代表二叉树的根， $L$  代表根结点的左子树， $R$  代表根结点的右子树。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

4. 下列二叉排序树中，满足平衡二叉树定义的是（ ）。



5. 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                      D. 119

6. 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

7. 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

8. 下列叙述不符合  $m$  阶 B 树定义要求的是 ( )。

- A. 根节点最多有  $m$  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

9. 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

10. 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. 起泡排序              B. 插入排序              C. 选择排序              D. 二路归并排序

41. (10 分) 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:

- 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  $v$ , 加入到最短路径中, 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, 请证明之; 否则, 请举例说明。

42. (15 分) 已知一个带有表头结点的单链表, 结点结构为: data link

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 data 域的值, 并返回 1; 否则, 只返回 0。要求:

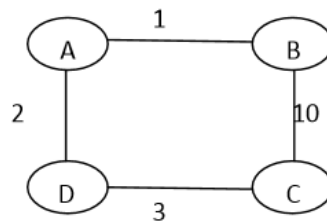
- (1) 描述算法的基本设计思想;
- (2) 描述算法的详细实现步骤;
- (3) 根据设计思想和实现步骤, 采用程序设计语言描述算法 (使用 C、C++ 或 Java 语言实现), 关键之处请给出简要注释。



- |      |      |      |      |       |
|------|------|------|------|-------|
| 1. B | 2. C | 3. D | 4. B | 5. C  |
| 6. B | 7. A | 8. D | 9. A | 10. B |

41. 解答:

该方法不一定能(或不能)求得最短路径。例如,对于下图所示的带权图,如果按照题中的原则,从A到C的最短路径是A->B->C,事实上其最短路径是A->D->C。



42. 解答:

(1) 算法的基本设计思想 (5分):

问题的关键是设计一个尽可能高效的算法,通过链表的一趟遍历,找到倒数第k个结点的位置。算法的基本设计思想是:定义两个指针变量p和q,初始时均指向头结点的下一个结点(链表的第一个结点)。p指针沿链表移动;当p指针移动到第k个结点时,q指针开始与p指针同步移动;当p指针移动到最后一个结点时,q指针所指示结点为倒数第k个结点。以上过程对链表仅进行一遍扫描。

(2) 算法的详细实现步骤 (5分):

- ①count=0, p和q指向链表表头结点的下一个结点;
- ②若p为空,转⑤;
- ③若count等于k,则q指向下一个结点;否则, count=count+1;
- ④p指向下一个结点,转②;

⑤若count等于k,则查找成功,输出该结点的data域的值,返回1;否则,说明k值超过了线性表的长度,查找失败,返回0;

⑥算法结束。

(3) 算法实现 (5分):

```

typedef int ElemType; // 链表数据的类型定义
typedef struct LNode { // 链表结点的结构定义
 ElemType data; // 结点数据
 struct LNode *link; // 结点链接指针
}*LinkList;

int Search_k(LinkList list, int k) {
 // 查找链表list倒数第k个结点,并输出该结点data域的值
 LinkList p = list->link, q = list->link; // 指针p、q指示第一个结点
 int count = 0;
 while (p != NULL) { // 遍历链表直到最后一个结点
 if (count < k) count++; // 计数,若count < k只移动p
 else q = q->link; p = p->link; // 之后让p、q同步移动
 }
}

```

```

} //while
if(count<k)return 0; //查找失败返回 0
else{//否则打印并返回 1
 printf(“%d”,q->data);
 return 1;
}
} //Search_k

```

提示：算法程序题，如果能够写出数据结构类型定义、正确的算法思想都会至少给一半以上分数，如果能用伪代码写出自然更好，比较复杂的地方可以直接用文字表达。

若所给出的算法采用一遍扫描方式就能得到正确结果，可给满分 15 分；若采用两遍或多遍扫描才能得到正确结果的，最高分 10 分。若采用递归算法得到正确结果的，最高给 10 分；若实现算法的空间复杂度过高（使用了大小与 k 有关的辅助数组），但结果正确，最高给 10 分。