硕士研究生入学考试初试专业课资料
计算机专业考研

统考真题——数据结构部分
（2009-2012）

2009 年计算机统考——数据结构部分

一、单项选择题：每小题 2 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，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。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．给定二叉树如图 A-1 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_____。

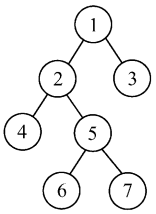
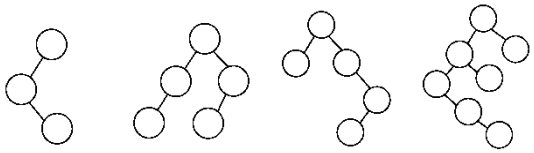


图　A-1

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_____。



A.　　B.　　C.　　D.

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．下列叙述中，不符合 m 阶 B 树定义要求的是_____。

A．根结点最多有 m 棵子树　　B．所有叶结点都在同一层上

C．各结点内关键字均升序或降序排列　　D．叶结点之间通过指针链接

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．冒泡排序　　B．插入排序　　C．选择排序　　D．二路归并排序

二、综合应用题

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，算法输出该结点的 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。

2009 年计算机统考——数据结构部分解析

一、单项选择题

1．**B**。考查栈和队列的特点及应用。

C 和 D 直接排除，缓冲区的特点需要先进先出，若用栈，先进入缓冲区的数据则要排队到最后才能打印，不符题意，故选 B。

2．**C**。考查栈的最大递归深度。

时刻注意栈的特点是先进后出。出入栈的详细过程见表 A-3。

表　A-3

序号	说明	栈内	栈外	序号	说明	栈内	栈外
1	a 入栈	a		8	e 入栈	ae	bdc
2	b 入栈	ab		9	f 入栈	aef	bdc
3	b 出栈	a	b	10	f 出栈	ae	bdcf
4	c 入栈	ac	b	11	e 出栈	a	bdcfe
5	d 入栈	acd	b	12	a 出栈		bdcfea
6	d 出栈	ac	bd	13	g 入栈	g	bdcfea
7	c 出栈	a	bdc	14	g 出栈		bdcfeag

栈内的最大深度为 3，故栈 S 的容量至少是 3。

3．**D**。考查二叉树的特殊遍历。

分析遍历后的结点序列，可以看出根结点是在中间被访问的，而右子树结点在左子树之前，得遍历的方法是 RNL。本题考查的遍历方法并不是二叉树的三种基本遍历方法，对于考生而言，重要的是要掌握遍历的思想。

4．**B**。考查平衡二叉树的定义。

根据平衡二叉树的定义有，任意结点的左、右子树高度差的绝对值不超过 1。而其余三个答案均可以找到不符合的结点。

5．**C**。考查完全二叉树的特点。

完全二叉树比满二叉树只是在最下面一层的右边缺少了部分叶结点，而最后一层之上是个满二叉树，并且只有最后两层有叶结点。第 6 层有叶结点则完全二叉树的高度可能为 6 或 7，显然树高为 7 时结点更多。若第 6 层上有 8 个叶结点，则前六层为满二叉树，而第 7 层缺失了 8×2=16 个叶结点，故完全二叉树的结点个数最多为 $2^7-1-16=111$ 个结点。

6．**B**。考查森林和二叉树的转换。

森林与二叉树的转换规则为“左孩子右兄弟”。在最后生成的二叉树中，父子关系在对应森林关系中可能是兄弟关系或原本就是父子关系。

情形Ⅰ：若结点 v 是结点 u 的第二个孩子结点，在转换时，结点 v 就变成结点 u 第一个孩子的右孩子，符合要求。

情形Ⅱ：结点 u 和 v 是兄弟结点的关系，但二者之中还有一个兄弟结点 k，则转换后，结点 v 就变为结点 k 的右孩子，而结点 k 则是结点 u 的右孩子，符合要求。

情形Ⅲ：结点 v 的父结点要么是原先的父结点或兄弟结点。若结点 u 的父结点与 v 的父结点是兄弟关系，则转换之后，不可能出现结点 u 是结点 v 的父结点的父结点。

7．**A**。考查无向连通图的特性。

每条边都连接了两个结点，则在计算顶点的度之和时，这条边都被计算了两次，故所有

顶点的度之和为边数的两倍，显然必为偶数。而Ⅱ和Ⅲ则不一定正确，如对顶点数 N≥1 无向完全图不存在一个顶点的度为 1，并且边数与顶点数的差要大于 1。

8．**D**。考查 m 阶 B-树的定义。

选项 A、B 和 C 都是 B-树的特点，而选项 D 则是 B+树的特点。注意区别 B-树和 B+树各自的特点。

9．**A**。考查小根堆的调整操作。

小根堆在逻辑上可以用完全二叉树来表示，根据关键序列得到的小顶堆的二叉树形式如图 A-4a 所示。

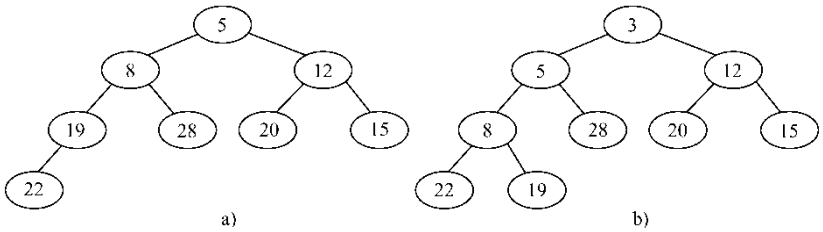


图　A-4

插入关键字 3 时，先将其放在小顶堆的末端，再将该关键字向上进行调整，得到的结果如图 A-4b 所示。所以，调整后的小顶堆序列为 3，5，12，8，28，20，15，22，19。

10．**B**。考查各排序算法的特点。

解答本题之前要对不同排序算法的特点极为清楚。对于冒泡排序和选择排序而言，每趟过后都能确定一个元素的最终位置，而由题目中所说，前两个元素和后两个元素均不是最小或最大的两个元素并按序排列。答案 D 中的二路归并排序，第一趟排序结束都可以得到若干个有序子序列，而此时的序列中并没有两两元素有序排列。插入排序在每趟排序结束后能保证前面的若干元素是有序的，而此时第二趟排序后，序列的前三个元素是有序的，符合其特点。

二、综合应用题

41．解答：

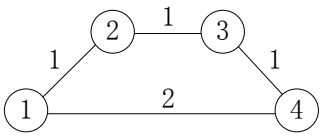


图　A-5

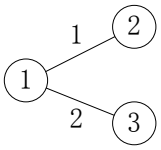


图　A-6

该方法不一定能（或不能）求得最短路径。（4 分）

举例说明：（6 分）

图 A-5 中，设初始顶点为 1，目标顶点为 4，欲求从顶点 1 到顶点 4 之间的最短路径，显然这两点之间的最短路径长度为 2。利用给定方法求得的路径长度为 3，但这条路径并不是这两点之间的最短路径。

图 A-6 中，设初始顶点为 1，目标顶点为 3，欲求从顶点 1 到顶点 3 之间的最短路径。利用给定的方法，无法求出顶点 1 到顶点 3 的路径。

【评分说明】

①若考生回答“能求得最短路径”，无论给出何种证明，均不给分。

②考生只要举出类似上述的一个反例说明“不能求得最短路径”或答案中体现了“局部

最优不等于全局最优”的思想，均可给 6 分；若举例说明不完全正确，可酌情给分。

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：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）：

① count=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；

④ p 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。

【评分说明】

①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，最高给 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；若实现的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。

②参考答案中只给出了使用 C 语言的版本，使用 C++/JAVA 语言正确实现的算法同样给分。

③若在算法基本思想描述和算法步骤描述中因文字表达没有非常清晰地反映出算法的思路，但在算法实现中能够清晰看出算法思想和步骤且正确，按照①的标准给分。

④若考生的答案中算法基本思想描述、算法步骤描述或算法实现中部分正确，可酌情给分。

2010 年计算机统考——数据结构部分

一、单项选择题

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．d c e b f a　　B．c b d a e f

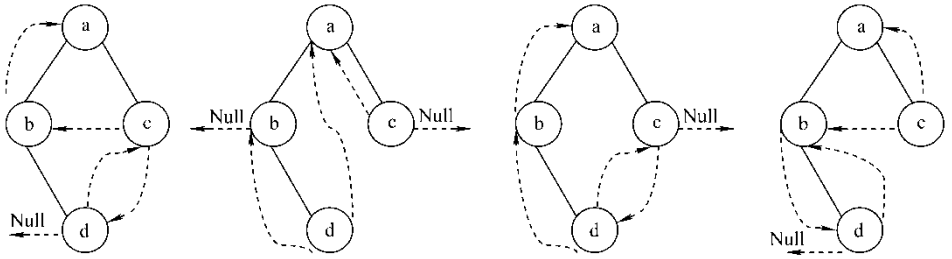
C．b c a e f d　　D．a f e d c b

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．b a c d e　　B．d b a c e

C．d b c a e　　D．e c b a d

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．在图 B-1 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_____。

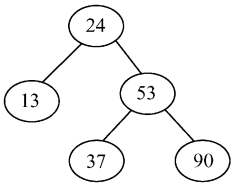


图 B-1

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≥2）个权值均不相同的字符构造成赫夫曼树。下列关于该赫夫曼树的叙述中，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．对图 B-2 进行拓扑排序，可以得到不同的拓扑序列的个数是_____。

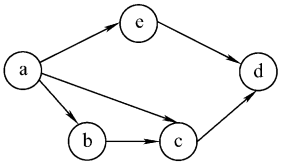


图　B-2

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的处理顺序无关

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．冒泡排序　　B．希尔排序　　C．归并排序　　D．基数排序

二、综合应用题

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(key)=(key×3) 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（X0, X1,…, Xn−1）变换为（Xp, Xp+1,…, Xn−1, X0, X1,…, Xp−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。

2010 年计算机统考——数据结构部分解析

一、单项选择题

1．**D**。考查限定条件的出栈序列。

A 可由 in，in，in，in，out，out，in，out，out，in，out，out 得到；

B 可由 in，in，in，out，out，in，out，out，in，out，in，out 得到；

C 可由 in，in，out，in，out，out，in，in，out，in，out，out 得到；

D 可由 in，out，in，in，in，in，in，out，out，out，out，out 得到，但题意要求不允许连续三次退栈操作，故 D 错。

2．**C**。考查受限的双端队列的出队序列。

A 可由左入，左入，右入，右入，右入得到；B 可由左入，左入，右入，左入，右入得到；D 可由左入，左入，左入，右入，左入得到。所以不可能得到 C。

3．**D**。考查线索二叉树的基本概念和构造。

题中所给二叉树的后序序列为 dbca。结点 d 无前驱和左子树，左链域空，无右子树，右链域指向其后继结点 b；结点 b 无左子树，左链域指向其前驱结点 d；结点 c 无左子树，左链域指向其前驱结点 b，无右子树，右链域指向其后继结点 a。

4．**C**。考查平衡二叉树的插入算法。

插入 48 以后，该二叉树根结点的平衡因子由−1 变为−2，失去平衡，需进行两次旋转（先右旋后左旋）操作。

5．**B**。考查树结点数的特性。

设树中度为 i（i=0，1，2，3，4）的结点数分别为 N_i，树中结点总数为 N，则树中各结点的度之和等于 N−1，即 $N=1+N_1+2N_2+3N_3+4N_4=N_0+N_1+N_2+N_3+N_4$，根据题设中的数据，即可得到 $N_0=82$，即树 T 的叶结点的个数是 82。

6．**A**。考查赫夫曼树的特性。

赫夫曼树为带权路径长度最小的二叉树，不一定是完全二叉树。赫夫曼树中没有度为 1 的结点，B 正确；构造赫夫曼树时，最先选取两个权值最小的结点作为左、右子树构造一棵新的二叉树，C 正确；赫夫曼树中任一非叶结点 P 的权值为其左、右子树根结点权值之和，其权值不小于其左、右子树根结点的权值，在与结点 P 的左、右子树根结点处于同一层的结点中，若存在权值大于结点 P 权值的结点 Q，那么结点 Q 的兄弟结点中权值较小的一个应该与结点 P 作为左、右子树构造新的二叉树。综上可知，赫夫曼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．**C**。考查图的连通性。

要保证无向图 G 在任何情况下都是连通的，即任意变动图 G 中的边，G 始终保持连通，首先需要 G 的任意 6 个结点构成完全连通子图 G1，需 15 条边，然后再添一条边将第 7 个结点与 G1 连接起来，共需 16 条边。

8．**B**。考查拓扑排序序列。

图 B-2 中有 3 个不同的拓扑排序序列，分别为 abced、abecd、aebcd。

9．**B**。考查折半查找的过程。

具有 n 个结点的判定树的高度为 $\lfloor \log_2 n \rfloor+1$，长度为 16，高度为 5，所以最多比较 5 次。

10．**D**。考查快速排序。

递归次数与各元素的初始排列有关。如果每一次划分后分区比较平衡，则递归次数少；如果划分后分区不平衡，则递归次数多。递归次数与处理顺序无关。

11．**A**。考查各种排序算法的过程。

看第一趟可知仅有 88 被移到最后。

如果是希尔排序，则 12，88，10 应变为 10，12，88。因此排除希尔排序。

如果是归并排序，则长度为 2 的子序列是有序的。因此可排除归并排序。

如果是基数排序，则 16，5，10 应变为 10，5，16。因此排除基数排序。

可以看到，每一趟都有一个元素移到其最终位置，符合冒泡排序特点。

二、综合应用题

41．解答：

（1）由装载因子为 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 0～9。所构造的散列函数值见表 B-3。

表　B-3

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，所构造的散列表见表 B-4。

表　B-4

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的，在等概率的情况下，各关键字的查找次数见表 B-5。

表　B-5

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

故，$ASL_{成功}$=查找次数/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 MOD 7，初始只可能在 0～6 的位置。等概率情况下，查找 0～6 位置查找失败的查找次数见表 B-6。

表　B-6

H(key)	0	1	2	3	4	5	6
次数	3	2	1	2	1	5	4

故，$ASL_{不成功}$=查找次数/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：

可以将这个问题看作是把数组 ab 转换成数组 ba（a 代表数组的前 p 个元素，b 代表数组中余下的 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）上述算法中 3 个 Reverse 函数的时间复杂度分别为 O(p/2)、O((n−p)/2)和 O(n/2)，故所设计的算法的时间复杂度为 O(n)，空间复杂度为 O(1)。

另解，借助辅助数组来实现。

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 O(n)，空间复杂度为 O(p)。

2011 年计算机统考——数据结构部分

一、单项选择题

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 分别指向队头元素和队尾元素。若初始时队列为空，且要求第 1 个进入队列的元素存储在 A[0]处，则初始时 front 和 rear 的值分别是_____。

A．0，0　　B．0，n−1　　C．n−1，0　　D．n−1，n−1

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

二、综合应用题

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	∞	∞	∞	5	∞	∞	∞	4	3	∞	∞	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≥1）的升序序列 S，处在第⌊L/2⌉个位置的数称为 S 的中位数。例如，若序列 S1=（11，13，15，17，19），则 S1 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 S2=（2，4，6，8，20），则 S1 和 S2 的中位数是 11。现在有两个等长升序序列 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。

2011 年计算机统考——数据结构部分解析

一、单项选择题

1．**A**。考查时间复杂度的计算。

在程序中，执行频率最高的语句为“x=2*x”。设该语句共执行了 t 次，则，$2^{t+1}=n/2$，故 $t=\log_2(n/2)-1=\log_2 n-2$，得 $T(n)=O(\log_2 n)$。

2．**B**。考查出栈序列。

出栈顺序必为 d_c_b_a_，e 的顺序不定，在任一个“_”上都有可能。

3．**B**。考查循环队列的性质。

入队时由于要执行(rear+1)%n 操作，所以如果入队后指针指向 0，则 rear 初值为 n-1，而由于第一个元素在 A[0]中，插入操作只改变 rear 指针，所以 front 为 0 不变。

4．**C**。考查完全二叉树的性质。

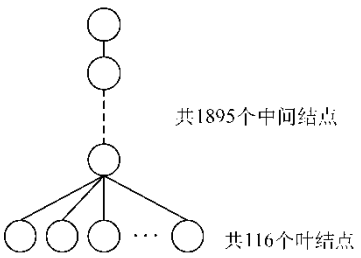
根据完全二叉树的性质，最后一个分支结点的序号为⌊768/2⌋=384，故叶子结点的个数为 768-384=384。

5．**C**。考查二叉树的遍历算法。

前序序列为 LRN，后序序列为 NLR，由于前序序列和后序序列刚好相反，故不可能存在一个结点同时存在左右孩子，即二叉树的高度为 4。1 为根结点，由于根结点只能有左孩子（或右孩子），因此，在中序序列中，1 或在序列首或在序列尾，ABCD 皆满足要求。仅考虑以 1 的孩子结点 2 为根结点的子树，它也只能有左孩子（或右孩子），因此，在中序序列中，2 或在序列首或序列尾，ABD 皆满足要求。

6．**D**。考查树和二叉树的转换。

树转换为二叉树时，树中每一个分支结点的所有子结点中的最右子结点无右孩子，根结点转换后也没有右孩子，因此，对应的二叉树中无右孩子的结点个数=分支结点数+1=2011-116+1=1896。通常本题应采用特殊法解，设题意中的树是如下图所示的结构，则对应的二叉树中仅有前 115 个叶结点有右孩子，故无右孩子的结点个数=2011-115=1896。



7．**A**。考查二叉排序树的查找过程。

在二叉排序树中，左子树结点值小于根结点，右子树结点值大于根结点。在选项 A 中，当查找到 91 后再向 24 查找，说明这一条路径（左子树）之后查找的数都要比 91 小，而后面却查找到了 94，因此错误。

8．**C**。考查图的基本概念。

回路对应于路径，简单回路对应于简单路径，故Ⅰ错误；稀疏图是边比较少的情况，此时用邻接矩阵必将浪费大量的空间，应该选用邻接表，故Ⅱ错误。存在回路的图不存在拓扑序列，故Ⅲ正确。

9．**D**。考查散列表的性质。

Hash 表的查找效率取决于：哈希函数、处理冲突的方法和装填因子。显然，冲突的产

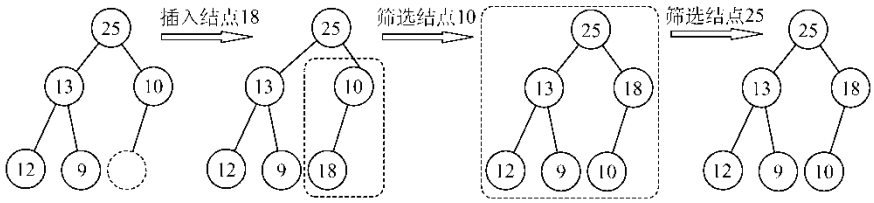
生概率与装填因子（即表中记录数与表长之比）的大小成正比，Ⅰ错误。冲突是不可避免的，但处理冲突的方法应避免非同义词之间地址的争夺，Ⅲ正确。

10．**A**。考查排序的基本特点。

对绝大部分内部排序而言，只适用于顺序存储结构。快速排序在排序的过程中，既要从后向前查找，也要从前向后查找，因此宜采用顺序存储。

11．**B**。考查堆的调整。

首先 18 与 10 比较，交换位置，再与 25 比较，不交换位置。共比较了 2 次，调整的过程如下图所示。



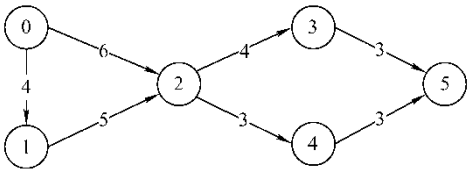
二、综合应用题

41．解答：

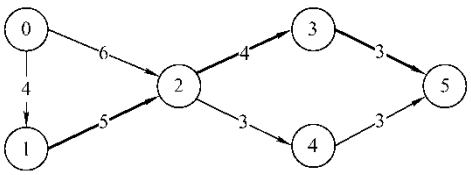
（1）用“平移”的思想，将前 5 个、后 4 个、后 3 个、后 2 个、后 1 个元素，分别移动到矩阵对角线（“0”）右边的行上。图 G 的邻接矩阵 $\boldsymbol{A}$ 如下所示。

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

（2）根据上面的邻接矩阵，画出有向带权图 G，如下图所示。



3）即寻找从 0 到 5 的最长路径。得到关键路径为 0→1→2→3→5（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。

③ 若 a>b，则舍弃序列 A 中较大的一半，同时舍弃序列 B 中较小的一半，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(int A[],int B[],int n){
    int s1=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])
            return A[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
            }
        }
    }
    return A[s1]<B[s2]? A[s1]:B[s2];
}
```

（3）算法的时间复杂度为 $O(\log_2 n)$，空间复杂度为 O(1)。

2012 年计算机统考——数据结构部分

一、单项选择题

1．求整数 n(n≥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‘+’、‘-’、‘*’、‘/’、‘(’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，若栈初始时为空，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_____。

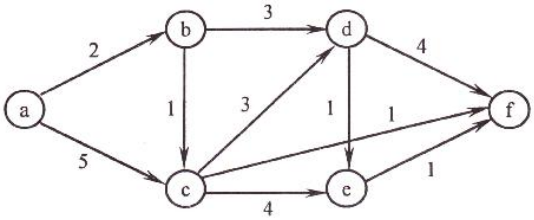
A．O(n)　　B．O(e)　　C．O(n+e)　　D．O(n*e)

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

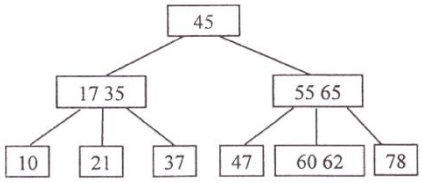
Ⅱ．所有权值最小的边一定会出现在所有的最小生成树中

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

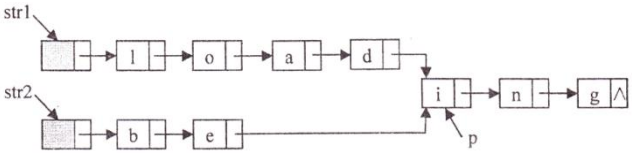
二、综合应用题

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≥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

，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 语音描述算法，关键之处给出注释。

（3）说明你所设计算法的时间复杂度。

2012 年计算机统考——数据结构部分解析

一、单项选择题

1．**B**。考查时间复杂度的计算。

该程序中使用了递归运算。本题中递归的边界条件是 n<=1，每调用一次 fact()，传入该层 fact()的参数值减 1（注意不是 n 减 1），因此执行频率最高的语句是 return n*fact(n-1)，一共执行了 n-1 次，而每一层 fact(i)运算只包含一次乘法。如果采用递归式来表示时间复杂度，则：

$$T(n)=\begin{cases} O(1) & n \le 1 \\ T(n-1)+1 & n > 1 \end{cases}$$

时间复杂度为 O(n)。

2．**A**。考查栈的应用、表达式求值。

表达式求值是栈的典型应用。通常涉及中缀表达式和后缀表达式。中缀表达式不仅依赖运算符的优先级，还要处理括号。后缀表达式的运算符在表达式的后面且没有括号，其形式已经包含了运算符的优先级。所以从中缀表达式转换到后缀表达式需要用运算符栈对中缀表达式进行处理，使其包含运算法优先级的信息，从而转换为后缀表达式的形式。转换过程如下表：

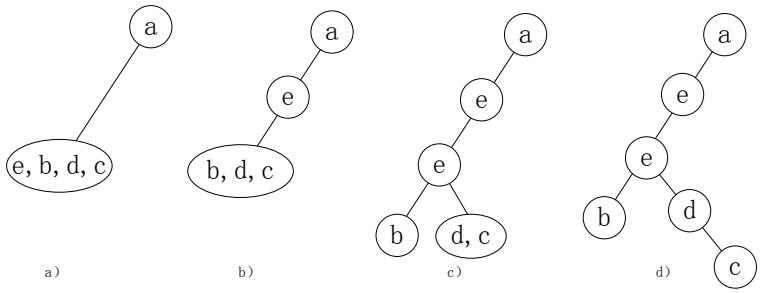
运算符栈	中缀未处理部分	后缀生成部分
#	a+b-a*((c+d)/e-f)+g	
#	+b-a*((c+d)/e-f)+g	a
+	b-a*((c+d)/e-f)+g	a
+	-a*((c+d)/e-f)+g	ab
-	a*((c+d)/e-f)+g	ab+
-	*((c+d)/e-f)+g	ab+a
-*	((c+d)/e-f)+g	ab+a
-*((	c+d)/e-f)+g	ab+a
-*((	+d)/e-f)+g	ab+ac
-*((+	d)/e-f)+g	ab+ac
-*((+	)/e-f)+g	ab+acd
-*(	/e-f)+g	ab+acd+
-*(/	e-f)+g	ab+acd+
-*(/	-f)+g	ab+acd+e
-*(-	f)+g	ab+acd+e/
-*(-	)+g	ab+acd+e/f
-*	+g	ab+acd+e/f-
-	+g	ab+acd+e/f-*
#	+g	ab+acd+e/f-*-
+	g	ab+acd+e/f-*-
#		ab+acd+e/f-*-g

3．**A**。考查树的遍历、及由遍历序列确定二叉树的树形。

前序序列和后序序列不能唯一确定一棵二叉树，但可以确定二叉树中结点的祖先关系：当两个结点的前序序列为 XY 与后序序列为 YX 时，则 X 为 Y 的祖先。考虑前序序列 a,e,b,d,c、

后序序列 b,c,d,e,a，可知 a 为根结点，e 为 a 的孩子结点；此外，a 的孩子结点的前序序列 e,b,d,c、后序序列 b,c,d,e，可知 e 是 bcd 的祖先，故根结点的孩子结点只有 e。本题答案为 A。

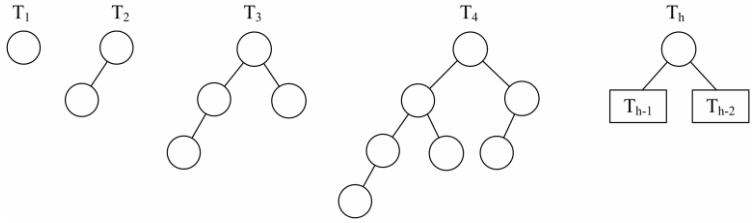
【特殊法】 前序序列和后序序列对应着多棵不同的二叉树树形，我们只需画出满足该条件的任一棵二叉树即可，任意一棵二叉树必定满足正确选项的要求。



显然选 A，最终得到的二叉树满足题设中前序序列和后序序列的要求。

4．**B**。考查平衡二叉树的最少结点情况。

所有非叶结点的平衡因子均为 1，即平衡二叉树满足平衡的最少结点情况，如图 xxx 所示。画图时，先画出 T_1 和 T_2；然后新建一个根结点，连接 T_2、T_1 构成 T_3；新建一个根结点，连接 T_3、T_2 构成 T_4；……依此类推，直到画出 T_6，可知 T_6 的结点数为 20。对于高度为 N 的题述的平衡二叉树，它的左、右子树分别为高度为 N-1 和 N-2 的所有非叶结点的平衡因子均为 1 的平衡二叉树。二叉树的结点总数公式为：$C_N=C_{N-1}+C_{N-2}+1$，$C_2=2$，$C_3=4$，递推可得 $C_6=20$。



【排除法】 对于选项 A，高度为 6、结点数为 10 的树怎么也无法达到平衡。对于选项 C，接点较多时，考虑较极端情形，即第 6 层只有最左叶子的完全二叉树刚好有 32 个结点，虽然满足平衡的条件，但显然再删去部分结点，依然不影响平衡，不是最少结点的情况。同理 D 错误。只可能选 B。

5．**C**。考查不同存储结构的图遍历算法的时间复杂度。

广度优先遍历需要借助队列实现。邻接表的结构包括：顶点表；边表（有向图为出边表）。当采用邻接表存储方式时，在对图进行广度优先遍历时每个顶点均需入队一次（顶点表遍历），故时间复杂度为 O(n)，在搜索所有顶点的邻接点的过程中，每条边至少访问一次（出边表遍历），故时间复杂度为 O(e)，算法总的时间复杂度为 O(n+e)。

6．**C**。考查拓扑排序、与存储结构和图性质的关系。

对角线以下元素均为零，表明该有向图是一个无环图，因此一定存在拓扑序列。对于拓扑序列是否唯一，我们试举一例：设有向图的邻接矩阵 $\begin{bmatrix}0&1&1\\0&0&0\\0&0&0\end{bmatrix}$，则存在两个拓扑序列。所以该图存在可能不唯一的拓扑排序。

结论：对于任一有向图，如果它的邻接矩阵中对角线以下（或以上）的元素均为零，则存在拓扑序列（可能不唯一）。反正，若图存在拓扑序列，却不一定能满足邻接矩阵中主对角线以下的元素均为零，但是可以通过适当地调整结点编号，使其邻接矩阵满足前述性质。

7．**C**。考查 Dijkstra 算法最单源最短路径。

从 a 到各顶点的最短路径的求解过程：

顶点	第 1 趟	第 2 趟	第 3 趟	第 4 趟	对 5 趟
b	(a,b) 2				
c	(a,c) 5	(a,b,c) 3			
d	∞	(a,b,d) 5	(a,b,d) 5	(a,b,d) 5	
e	∞	∞	(a,b,c,f) 4		
f	∞	∞	(a,b,c,e) 7	(a,b,c,e) 7	(a,b,d,e) 6
集合 S	{a,b}	{a,b,c}	{a,b,c,f}	{a,b,c,f,d}	{a,b,c,f,d,e}

后续目标顶点依次为 f,d,e，

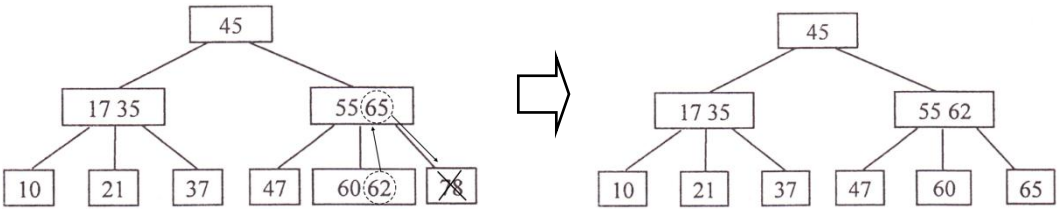
【排除法】 对于 A，若下一个顶点为 d，路径 a,b,d 的长度 5，而 a,b,c,f 的长度仅为 4，显然错误。同理可以排除 B。将 f 加入集合 S 后，采用上述的方法也可以排除 D。

8．**A**。考查最小生成树、及最小生成树算法的性质。

对于 I，最小生成树的树形可能不唯一（这是因为可能存在权值相同的边），但是代价一定是唯一的，I 正确。对于 II，如果权值最小的边有多条并且构成环状，则总有权值最小的边将不出现在某棵最小生成树中，II 错误。对于 III，设 N 个结点构成环，N-1 条边权值相等，则从不同的顶点开始普里姆算法会得到 N-1 中不同的最小生成树，III 错误。对于 IV，当最小生成树唯一时（各边的权值不同），普里姆算法和克鲁斯卡尔算法得到的最小生成树相同，IV 错误。

9．**D**。考查 B-树的删除操作。

对于上图所示的 3 阶 B-树，被删关键字 78 所在结点在删除前的关键字个数=1=⌈3/2⌉-1，且其左兄弟结点的关键字个数=2≥⌈3/2⌉，属于“兄弟够借”的情况，则需把该结点的左兄弟结点中最大的关键字上移到双亲结点中，同时把双亲结点中大于上移关键字的关键字下移到要删除关键字的结点中，这样就达到了新的平衡，如下图所示。



10．**A**。考查各种内部排序算法的性质。

对于Ⅰ，简单选择排序每次选择未排序列中的最小元素放入其最终位置。对于Ⅱ，希尔排序每次是对划分的子表进行排序，得到局部有序的结果，所以不能保证每一趟排序结束都能确定一个元素的最终位置。对于Ⅲ，快速排序每一趟排序结束后都将枢轴元素放到最终位置。对于Ⅳ，堆排序属于选择排序，每次都将大根堆的根结点与表尾结点交换，确定其最终位置。对于Ⅴ，二路归并排序每趟对子表进行两两归并从而得到若干个局部有序的结果，但无法确定最终位置。

11．**D**。考查折半插入和直接插入的区别。

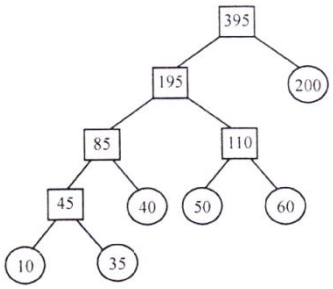
折半插入排序与直接插入排序都是将待插入元素插入前面的有序子表，区别是：确定当前记录在前面有序子表中的位置时，直接插入排序是采用顺序查找法，而折半插入排序是采用折半查找法。排序的总趟数取决于元素个数 n，两者都是 n-1 趟。元素的移动次数都取决于初试序列，两者相同。使用辅助空间的数量也都是 O(1)。折半插入排序的比较次数与序列初态无关，为 $O(n\log_2 n)$；而直接插入排序的比较次数与序列初态有关，为 $O(n)\sim O(n^2)$。

二、综合应用题

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，比较的总次数最多为：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>=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(LinkList str1,LinkList str2){
    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。