上海大学2015~2016数据结构(一)

一、填空

1、在以HL为表头指针的带表头附加结点的单链表和循环单链表中，判断链表为空的条件分别为 **HL→next =NULL；和 HL=HL→next；**。

2、调整得到的最小堆序列的最后一个元素为（**17**）

> 参考：https://blog.csdn.net/hrn1216/article/details/51465270

3、对于一个长读为n的顺序存储的线性表，在表头插入元素的时间复杂度为（**O(n)** ）。

> 对于一个长度为n的顺序存储的线性表，在表头插入元素的时间复杂度为0(n)，在表尾插入元素的时间复杂度为0(1)。
>
> 顺序存储的线性表，是用数组实现的。在表尾插入元素，只要直接在表尾增加一个元素，并修改表的元素个数（加1）。所以其复杂度为0(1)。

拓展：

> 对于一个长度为n的单链存储的线性表，在表头插入元素的时间复杂度为__O(1）__，在表尾插入元素的时间复杂度为__O(n)__。
>
> 表头插入时间复杂度O(1），因为不需用移动元素，常数时间完成操作；表尾插入复杂度O(n)，因为每次操作都需用把指针先移动到表尾，需用n次移动。

4、设栈S和队列Q的初始状态均为空，元素abcdefg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bdcfeag，则栈S的容量至少是（**3**）

> 栈后入先出的特点。队列先入先出，即队列也是bdcfeag
>
> 开始a先入栈，然后b入栈，然后b出栈，队列b 然后c,d依次入栈，然后d,c依次出栈，队列bcd 然后e,f依次入栈，然后f e 出栈队列bdcfe
>
> a出栈，g进栈g出栈；
>
> 栈最多时为3 即 acd 、aef时栈深度为3

5、一个栈的入栈序列为1,2,3,...,n ，其出栈序列是 p1, p2, p3, ... pn 。若p2 = 3，则 p3 可能取值的个数是（**n-1**）

> 一个栈的入栈序列为1,2,3,.....n",其出栈序列是P1,P2,P3,P4...Pn,。若P2=3,P3可能取值的个数
>
> 根据栈先进后出,后进先出的特性,若P2=3,意味着出栈的第二位即为3.
>
> 那么整个栈可以3为节点,分为两个序列,{1,2}和{4,...n}
>
> P1可能为1,2,也可能为4,
>
> P3也可能为1,2,这里要注意1,2虽在3之前,但也能在P3出栈
>
> 而P3唯一取不到的值即已确定出栈的3,因此可取值的个数为n-1

> P3在P2之后出栈，所以除了3本身以外，其他的值均可以取到，因此可能取值的个数为n-1。

6、设以数组Q.elems[maxSize]存储循环队列的元素，同时以Q.rear和Q.length分别指示循环队列中的队尾位置和队列中所含元素的个数

```
Q.length==0      //队空的条件
Q.length==maxSize    //队满的条件
//入队
Q.rear=(Q.rear+1)%maxSize;// 移动指针
Q.elems[Q.rear]=x;// 入队元素
Q.length++;// 队列长度加1
```


> 参考：https://blog.csdn.net/cnds123321/article/details/106740109

7、一维数组的逻辑结构是(线性结构)，存储结构是(顺序结构)；对于二维或多维数组，分(以行为主序)和(以列为主序)两种不同存储方式。

> 一维数组是 顺序存储结构 二维数组或多维数组 行优先或者列优先 逻辑结构 线性结构

8、一个右子树为空的二叉树在后序线索化后，其空指针域的个数为：**2**

拓展：

> 一棵左子树为空的二叉树在先序线索化后，其中空的链域的个数是***D***。
>
> A．不确定
> B．0
> C．1
> D．2
>
> 解释：
>
> 左子树为空的二叉树的根结点的左线索为空(无前驱)，先序序列的最后结点的右线索为空(无后继)，共2个空链域。

9、设n 为哈夫曼树的叶子结点数目，则该哈夫曼树共有（**2n-1**）个结点。

> 首先，哈夫曼树是一个二叉树；第二点，哈弗曼树的度只有两种情况，一是只有两个度的结点，二是没有度的结点，即叶子结点。设两个度的节点数为n2, 已知叶子结点为n，总节点数减1等于分支数，可知n2+n-1=2*n2

——

> 如果这道题目里面的哈夫曼树是指二叉的话，那么答案是B，如果不确定是几叉的话，那么是A。
>
> 无论哈夫曼树是几叉，其特点是一致的（假设为m叉），即树中只存在度为0的结点（即叶结点）和度为m的结点。不妨设度为0的结点个数为x，度为m的结点个数为y，则存在一个等式x+y=my+1，即x=(m-1)y+1，x+y是树的总结点个数。
>
> 就这道题来说，假设哈夫曼树是二叉的话，则度为0的结点个数为N，度为2的结点个数为N-1，则结点总数为2N-1。

二、单选题

1、广义表A=(a),则表尾为 (**C　空表**)

2、若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算，则利用(**A　顺序表**)存储方式最节省时间

> “存取任一指定序号”最好的方法是实现“随机存取”，则可采用顺序表。并且，因为插入和删除操作都是在最后进行的，所以无需大量移动数据元素，选项A是最合适的。

3、假设栈初始为空，将中缀表达式**a/b+(c*d-e*f)/g**转化为等价的狗追表达式的过程中，当扫描到f时，栈中的元素依次是（**B　+　(-***）

> 参考：https://blog.csdn.net/qq_22073849/article/details/78416135

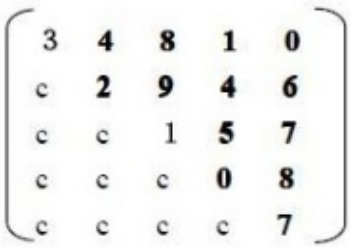
4、已知循环队列存储在一维数组A[0...n-1]，且队列非空时，front和rear分别指向队头元素和队尾元素。若初始时队列为空，且要求第一个进入队列的元素存储在A[0]处，则初始时front和rear的值分别是（**B. 0，n-1** ）

> 参考：https://blog.csdn.net/qq_22073849/article/details/78362502
>
> https://www.nowcoder.com/questionTerminal/1d83ef24b318444e81dd426c657bd61c

> 插入时，队头指针不变，队尾指针后移一位。该题约定队列非空时 front 和 rear 分别指向队头和队尾元素，即插入第一个元素在下标为0的位置后，队头队尾指针皆指向A[0]，此时可以反推出插入前，队头指针仍旧指向下标0，而队尾指针应指向前一位，也就是下标n-1的位置。注意，这道题的约定与大多数题约定队列为空时front=rear=0是不一样的约定，都是根据题意解题。

5、设有一个10阶的下三角矩阵A（包括对角线），按照从上到下、从左到右的顺序存储到连续的55个存储单元中，每个数组元素占1个字节的存储空间，则A[5][4]地址与A[0][0]的地址之差为（ B 19 ）。

(a) 下三角矩阵



(b) 上三角矩阵

下三角矩阵如图所示。如果A[0][0]的位置是1的话，不难理解A[5][4]的位置是：1+2+3+4+5+5 = 20。

地址差为19。

6、广义表A=（x, (a ,b)), (x,(a,b),y)), 则运算Head(Head(Tail(A)))结果为(**A x**)

题目中给出的：

A=(x, (a ,b)), (x, (a,b) ,y)) 少了一个括号配对，应该为：**A=（x, (a,b)) ,(x, (a,b), y))**

Head(Head(Tail(A)))) 多了一个括号，应该改为：**Head(Head(Tail(A)))**

由内到外依次运算：Head广义表的第一个元素，Tail取广义表除了第一个元素外的其他元素

1. Tail(A)= （x, (a,b), y)
2. Head(Tail(A))=(x)
3. Head(Head(Tail(A)))=(x)

7、函数substr("DATASTURCTURE",5,9)的返回值为(**A STRURCTURE**)

substr("DATASTURCTURE",5,9)
这个函数的意思是从字符串“DATASTURCTURE”的第五个字符S开始，截取后面的9个字符。
所以为：
STRURCTURE

8、

9、一棵完全二叉树上有1001个结点，其中叶子结点的个数是（**A 501**）

这棵完全--X．树的高度为[*]根据二叉树的性质，从第1层到第 9层共有结点2^9-1 = 511个。第10层全部是叶子结点，因此处于第10层的叶子结点数为 1001-511 = 490。同时注意到，第9层有2^{9-1}-490/2 = 11个叶子结点。因此共有490+11 =501个叶子结点。
也可以用另外一种方法来做。设二叉树的总结点数为n，叶子结点数为n_0，度为1的结点数为n_1，度为2的结点数为n_2，根据二叉树的性质有：$n_0=n_2+1$，$n=n_1+2n_2+1$，于是可得，$n=n_1+2n_0-1$，由于在完全二叉树中，度为1的结点总数n_1要么为0要么为1，此题中显然为0，这样才能保证等式两边都是奇数，因此$1001 = 2n_0-1$，解得$n_0 = 501$。

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

中序遍历时，先访问左子树，再访问根结点。n在m前，则n必须在m的左子树中。因此本题答案为C。

> 中序遍历为左根右，所以n一定要在m左边

> 参考：https://blog.csdn.net/qq_43839907/article/details/103359858

三、是非题

1、线性表采用链表存储时，查找第i个元素的时间与i的值无关 （×）

> 线性链表的存储是非顺序存储的，访问第i个元素是需要从第一个元素开始一个一个的来寻找，所以跟i值成正比，线性表在顺序存储时，是顺序存储的，访问第i个元素是可以直接访问到，时间复杂度为o(1)，与i无关

2、非空的双向循环链表中任何结点的前驱指针均不为空。（√）

> 是正确的。 只要是循环链表，任一一个节点的前驱指针和后继指针都不会为空。 双向循环链表是循环链表的一种，所以也适用于这个规律。
> 原因如下：
> 1 对于单向链表，是从第一个节点开始，到最后一个节点结束，其指向为
> P1->P2->P3->...->Pn
> 第一个节点P1的前驱指针和最后一个节点Pn的后继指针为空。
> 2 对于循环链表， 会将最后一个节点指向第一个节点，构成循环：
> P1->P2->P3->...->Pn->P1
> 而双向循环链表则是每个节点两个指针，分别指向上一个和下一个：
> P1<->P2<->P3<->...<->Pn<->P1\
> 从这个结构可以看出， 每一个节点的前驱和后继都不可能为空， 当只有一个节点的时候，前驱和后继都是自身。

3、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，则不可能得到的顺序是dbcae(√)

> 正确答案是 C
>
> 本题的队列实际上是一个输出受限的双端队列。A操作：a左入（或右入）、b左入、c右入、d右入、e右入。B操作：a左入（或右入）、b左入、c右入、d左入、e右入。D操作：a左入（或右入）、b左入、c左入、d右入、e左入。C操作：a左入（或右入）、b右入、因d未出，此时只能进队，c怎么进都不可能在b和a之间。
>
> 【另解】初始时队列为空，第1个元素a左入（或右入），而第2个元素b无论是左入还是右入都必与a相邻，而选项D中a与b不相邻，不合题意。

> [解析] 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。

4、用不带头结点的单链表存储队列时，队头指针指向队头结点，队尾指针指向队尾结点，则在进行出队操作时只需要修改队头指针(×)。

> 如果当前队列中仅有一个元素，则删除它时队头、队尾指针都需要修改。

5、稀疏矩阵压缩存储后，必然会失去随机存取功能（√）

> 稀疏矩阵压缩存储后，必会失去随机存取功能。稀疏矩阵在采用压缩存储后将会失去随机存储的功能。因为在这种矩阵中，非零元素的分布是没有规律的，为了压缩存储，就将每一个非零元素的值和它所在的行、列号做为一个结点存放在一起，这样的结点组成的线性表中叫三元组表，它已不是简单的向量，所以无法用下标直接存取矩阵中的元素。

> 稀疏矩阵用三元组存储，已经不能用简单的下标来实现访问，所以已经失去了随机访问的功能

6、广义表的组成元素可以是不同形式的元素（√）

7、一个广义表的表头为空表,则此广义表亦为空表（×）

> 举一个简单的反例就可说明为什么了,例如广义表L = { { },1,2 },该广义表的长度为3,深度为2,表头head(L) = { },tail(L) = {1,2},可以看出表头为{ },它是一个空表,但表L并不是空表的呀.

8、对一棵二叉树进行层次遍历时,应借助于一个队列(√)

> 应该借助于队列，二叉树的先序，后序，中序的非递归遍历才需要栈

> 由层次遍历的定义可知，在进行层次遍历时，对一层的结点访问完后，在按照他们的访问次序依次对各个节点的左右孩子顺序访问，这样一层一层的进行，先遇到的结点先访问，这与队列的操作原则比较吻合，因此在进行层次遍历时，可设置一个队列结构，遍历从二叉树的根节点开始，首先将根节点指针入队列，依次执行下面操作：
>
> 1）队列不空，出队列，去队头元素
>
> 2）访问该元素所指结点
>
> 3）若该元素所指结点的左右孩子结点非空，则该元素所指结点的左孩子指针和右孩子指针顺序入队。

9、二叉树中序线索化后，不存在空指针域（×）

> 非空二叉树中序遍历第一个结点无前驱，最后一个结点无后继，这两个结点的前驱线索和后继线索为空指针。

10、完全二叉树中，若一个结点没有左孩子，则它必是树叶（√）

> 完全二叉树如果没有左结点，则一定没有右结点，即没有左孩子，它就一定是树叶。

> 参考：https://blog.csdn.net/weixin_46678290/article/details/105408072

四、应用题

1、KMP算法，计算失效函数值，填写KMP算法每一轮的匹配过程

> 参考：https://blog.csdn.net/lemon_sun/article/details/37655509

2、已知频率，构造并画出哈夫曼树，求哈夫曼编码

五、问答题

1、已知一棵二叉树的层次序列、中序序列，画出该二叉树，写出后序序列，求高度

> 参考：https://bbs.csdn.net/topics/70107860

2、利用栈S1和S2模拟一个队列，完成入队、出队、判空操作

> 参考：https://blog.csdn.net/fan9511/article/details/38015891
>
> https://blog.csdn.net/weixin_44279771/article/details/107860898

六、算法填空题

1、将不带头结点的非空单链表A分解成两个单链表A和B

> 参考：https://blog.csdn.net/qq_43648337/article/details/107494392

2、在二叉树的二叉链表类模板中，实现非递归中序遍历二叉树

> 参考：http://c.biancheng.net/view/3389.html
>
> https://blog.csdn.net/qq_33060405/article/details/78505347

七、算法设计题

1、在单链表类中增加一个用于判断带头结点的单链表中数据元素是否是递增的成员函数

上海大学2014~2015数据结构(一)

一、填空题

1、带头结点的循环链表 L 中只有一个元素结点的条件是（**head->next->next=head**）。

拓展：

> **假设带头结点的单向循环链表的头指针为head,则该链表为空的判定条件是（ ）**
>
> 答：head–>next= =head

2、在顺序表中访问任意一结点的时间复杂度均为（**O（1）**），因此，顺序表也称为（**随机存取**）的数据结构

3、设栈S和队列Q的初始状态为空。元素a、b、c、d、e、f依次通过栈S，并且一个元素出栈后即进入队列Q，若出队的顺序为b、d、c、f、e、a，则栈S的容量至少应该为（**3**）

> [分析] 由于队列是先进先出线性表，队列Q的出队顺序为b、d、c、f、e、a，则入队顺序必定也是b、d、c、f、e、a，这一顺序就是栈S的出栈顺序。又由于入栈顺序为a、b、c、d、e、f，因此入栈和出栈顺序是：a、b入栈，b出栈，c、d入栈，d、c出栈、e、f入栈，f、e、a出栈，因此栈中驻留元素最多是3个，因此栈S的容量至少应该为3。

5、假设一棵二叉树的结点个数为1638，则它的最小高度是（**11** ）

> 高度最少的时候，就是除了叶子结点外，其他的每层都是满节点，即满二叉树。满二叉数节点和高度的关系有：节点数=2的高度次方-1，由此可以推算出，当节点数是1638的时候，该二叉树的最小高度是11层

6、在一个最小堆中，所有结点中的最大值一定在（**叶子**）结点中

拓展：

> 2．在一个最小堆中，堆顶结点的值是所有结点中的_最小值_____，具有最大值的元素可能在___叶子结点____，在一个最大堆中，堆顶结点的值是所有结点中的___最大值____。

7、广义表运算式 Tail(((a,b),(c,d)))的操作结果是（ **((c,d))** ）

> 表尾是指除了表头以外所有原子和子表组成的新表。
>
> 原表：((a,b),(c,d))
>
> 表头：(a,b)
>
> 表尾：((c,d)) // 表尾是新表所以多一层括号

8、已知串S='aaab',其Next数组值为（ **0 1 2 3** ）

> 序号：1 2 3 4
> 数组：a a a b
> next： 0 1 2 3
>
> 注意上边序号、数组和next的对应关系
>
> 求next值的过程：
> 前两位：next数组值前两位一定为01，即aaab中的前两位aa对应01，如上表中next第1，2位为0和1.其实这就可以选出答案了.
>
> 第三位：3a前面是2a（2a表示序号为2的a），2a的next数组值为1，找到序号为1的字符, 即1a,将2a和1a相比，两者相同，都是a，则3a的next值为2a的next值加1，即2；
>
> 第四位：4b前3a的next为2，找到序号为2的字符, 即2a, 将3a与2a相比，二者相同，则其next值为3a的next加1，为3.
> 结果为0123

> KMP算法的关键是计算一个next数组，也叫做失效函数

9、求哈夫曼树的带权路径长度

> 参考：
>
> https://blog.csdn.net/qq_33369979/article/details/100060262

> https://blog.csdn.net/xueba8/article/details/78477892

10、数据结构主要研讨数据的**逻辑结构**和**物理结构**，以及它们之间的**相互关系**，并对这种结构定义相应**操作/运算**，设计出相应的**算法**。

二、选择题

1、某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用（**A 仅有尾指针的单循环链表**）存储方式最节省运算时间。

> 仅有尾指针的单循环链表，可以非常方便地找到尾结点，尾结点后面的第一个结点往往是头结点，头结点的下一个结点就是第线性表的第一个结点。对最后一个元素和第一个元素操作对带尾指针的单循环链表是非常方便的。

2、在双向链表中，在p结点之后插入结点q的操作是(**B　q->next=p->next；p->next->prior=q；p->next=q；q->prior=p；**)。

> [解析] 在链表中，对指针的修改必须保持线性表的逻辑关系，否则，将违背线性表的逻辑特征。本题主要考查双向链表的插入算法中的指针的变化过程。虽然4个选项中的语句相同，但顺序不同，根据双向链表的结构特点可知选项B的操作顺序是正确的，其他3个选项的指针修改顺序不能完成在*p结点之后插入结点*q的操作。

3、为了增加内存空间的利用率和减少溢出的可能性，由两个栈共享一片连续的内存空间时，应将两个栈的栈底分别设在这片内存空间的两端。如此只有当(**C 两个栈的栈顶在栈空间的某一位置相遇**)时，才产生溢出。

> [解析] 在一个程序中需要同时使用具有相同成分类型的两个栈时，为避免造成存储空间的浪费，应采用双进栈操作。为两个栈共同开辟一个连续的存储空间，一个栈的栈底为该空间的始端，另一个栈的栈底为该存储空间的末端。当元素进栈时都从此存储空间的两端向中间"延伸"。只有当两个栈的栈顶在该存储空间的某处相遇时，才会发生上溢。

4、用不带头结点的单链表存储队列时，其队头指针指向队头结点，其队尾指针指向队尾结点，则在进行删除操作时(**D 队头、队尾指针都可能要修改**)。

> 如果当前队列中仅有一个元素，则删除它时队头、队尾指针都需要修改。
>
> 因为当队列中只有一个元素时，删除此元素后要将队列置空，此时要修改队尾指针，使尾指针与头指针相等（即Q.rear = Q.front）

5、稀疏矩阵采用压缩存储的目的主要是为了 （**C 节省存储空间**）。

> 对稀疏矩阵，目前还没有一个明确的定义，但是一般认为，稀疏矩阵是非零元素较零元素少，且分布没有一定规律的矩阵。在矩阵运算中和矩阵输入输出中，最方便的存储方式就是二维数组，对矩阵进行压缩不能简化矩阵运算，对输入输出也不能提供便利，而降低运算的时间复杂度主要与算法有关，一般对矩阵压缩后其运算的复杂度会增加。所以答案是节省存储空间。

6、设有一个10阶的对称矩阵A，采用压缩存储方式，以行序为主存储，a1,1为第一元素，其存储地址为1，每个元素占一个地址空间，则a8,5的地址是（**B　33**）。

> [解析] 这里数组下标从1开始，只存储其下三角形元素，在a8,5的前面有7行，第1行有1个元素，第2行有2个元素，...，第7行有7个元素，这7行共有(1+7)×7/2=28个元素，在第8行中，a8,5的前面有4个元素，所以，a8,5前有28+4=32个元素，其地址为33。

7、数据在计算机存储器内表示时，物理地址与逻辑地址不相同的，称之为（**C　链式存储结构**）

8、若一棵二叉树具有10个度为2的结点，5个度为1的结点，则二叉树的结点个数是(**C 26**)。

> 拓展：
>
> 若一棵二叉树具有10个度为2的结点， 5个度为1的结点，则度为0的结点 （即叶子结点）个数是（39） 。
>
> n=n0+n1+n2　 n=1+n1+2*n2 （n为结点总数，n0为度为0的结点数，n1为度为1的结点数，n2为度为2的结点数） 可以推出 n0=n2+1 因此，度为0的结点个数=10+1=11

9、下列有关二叉树的说法中，正确的是(**B．一棵二叉树的度可以小于2**)。

> [解析]二叉树的定义为：二叉树是结点的有限集合，这个有限集合或者为空集，或者由一个根结点及两棵不相交的分别称做这个根的左子树和右子树的二叉树所构成，这里的左子树和右子树也符合二叉树的定义。由二叉树定义可得到这些信息：二叉树可以是空集，当二叉树为空集时，度为0；每个结点有两棵可以是空集的子树，当一棵子树为空，另一棵子树来为空时，该结点的度为1，都不为空时，度为2。由此可判断A、C、D是错误的，二叉树的度可以为0、1，结点的度也可以是0、1；选项B是正确的。

10、一棵非空的二叉树的先序遍历序列与后序遍历序列正好相反，则该二叉树一定满足（**C　只有一个叶子结点**）

> 之所以不能选AB，是因为题目中只交代了非空，
>
> 而如果该二叉树只有根节点，即满足了AB条件，
>
> 而显然此状态下，三种遍历顺序的结果都相同。
>
> 因此我们只能选择C
>
> 先序：根->左->右
>
> 后序：左->右->根
>
> 只有两个节点是，可以省略为：
>
> 先序：根->孩子
>
> 后序：孩子->根
>
> 因此选C

三、是非题

1、线性表采用顺序存储结构时，可以进行较快的插入和删除。　　（×）

> [解析] 在顺序表上做插入删除，将引起大量元素的移动。散列表也不便于删除，只有链接表既便于插入删除(因为只需要修改指针，不需要移动元素)，又能(通过指针域)反应元素之间的逻辑关系。

2、单链表中取第i个元素的时间与i成正比。（√）

3、循环顺序队列不存在上溢问题　（×）

> **① "下溢"现象**
> 当队列为空时，做出队运算产生的溢出现象。“下溢”是正常现象，常用作程序控制转移的条件。
> **② "真上溢"现象**
> 当队列满时，做进栈运算产生空间溢出的现象。“真上溢”是一种出错状态，应设法避免。
> **③ "假上溢"现象**
> 由于入队和出队操作中，头尾指针只增加不减小，致使被删元素的空间永远无法重新利用。当队列中实际的元素个数远远小于向量空间的规模时，也可能由于尾指针已超越向量空间的上界而不能做入队操作。该现象称为"假上溢"现象。

> 参考：http://blog.sina.com.cn/s/blog_5b9734c501015wjp.html

4、若输入序列为abcd，则通过一个栈可以输出cdab　（×）

5、若一个广义表的表头为空表,则此广义表一定不为空表　（×）　类似的题目

举一个简单的反例就可说明为什么了,例如广义表L = { { },1,2 },该广义表的长度为3,深度为2,表头head(L) = { },tail(L) = {1,2},可以看出表头为{ },它是一个空表,但表L并不是空表的呀.

6、一个稀疏矩阵 A m n 采用三元组形式表示，若把三元组中有关行下标与列下标的值互换，并把m和n的值互换，就完成了A mn 的转置运算。（×）

> 行列互换，再按行排序

7、数据元素是数据的最小单位。　（×）

> 数据元素是数据的基本单位, 数据项是不可分割的最小单位
> 数据项是构成数据元素的最小单位

8、二叉树的存储结构必须采用二叉链表结构（×）

9、二叉树的第i层上一定有 2^(i-1)个结点。　（×）

> 二叉树的第i层上最多有 2^(i-1)个结点。

10、健壮的算法不会因非法的输入数据而出现莫名其妙的状态。　（√）

> 健壮的算法，当有非法输入的时候，程序会自动给出错误提示而不会出现莫名奇怪的状态

四、设计题

1、

【解答】操作步骤为：

① 将所有元素出栈并入队；

② 依次将队列元素出队，如果是偶数结点，则再入队，如果是奇数结点，则入栈；

③ 将奇数结点出栈并入队；

④ 将偶数结点出队并入栈；

⑤ 将所有元素出栈并入队；

⑥ 将所有元素出队并入栈即为所求。

> 参考：https://wenku.baidu.com/view/0cabf3ea81c758f5f61f677a.html

2、将稀疏矩阵的三元组顺序表修改为 带行指针数组的二元组顺序表

五、问答题

1、已知二叉树的先序、中序序列，求其高度、叶子结点数，画出该二叉树和后续线索

2、已知广义表，求其长度、深度、表头、表尾，画出其链式存储结构的示意图

六、算法填空题

1、单链表的原地逆转

> 参考：https://www.cnblogs.com/young-for-you/p/7286905.html
>
> http://blog.sina.com.cn/s/blog_68c53ea90101bcl5.html

2、在中序线索二叉树中插入左孩子

> 参考：https://wenku.baidu.com/view/0b085eec58eef8c75fbfc77da26925c52dc5910d.html

七、算法设计题

1、用不带头结点的单循环链表表示队列，完成入队、出队函数

> 参考：https://blog.csdn.net/weixin_44790790/article/details/110297513

19-朝承恩