1. 什么是ADT？它包含哪几个部分？有什么特点？
2. 写出以下各语句执行次数（左边括号内数字为语句序号）

(1) for (i=0;i<n;i++)

(2) for (j=1;j<n;j++)

(3) { c[i,j] = 0;

(4) for (k=1;k<n;k++)

(5) c[i,j]=c[i,j]+a[i,k]\*b[k,j]

}

1. j = ?

j = 0;

for( i=1; i<=n; i++)

for (j=1; j<=i; j++)

for (k=1; k<=j; k++)

j ++;

1. 语句@的频度?

x=91; y=100;

while (y>0)

@ if (x>100)

{ x-=10; y--; }

else x++;

1. 若长度为 n 的线性表采用顺序存储结构，在其第 i (1 ≤ i ≤ n+1) 个位置之后插入一个新元素的算法时间复杂度是多少？
2. 若采用带尾指针的单链表结构，则删除最后一个节点和第一个节点的复杂度分别为多少？
3. 已知长度为n的顺序表A=(a1, a2, …, an)。请将A转换为A’=(an, an-1, …, a1)。
4. 已知长度为n的单链表表A=(a1, a2, …, an)。请将A转换为A’=(an, an-1, …, a1)。
5. 已知线性表中的元素以值递增有序排列，并以顺序表为存储结构。试写一高效算法，删除表中所有值大于mink且小于maxk的元素 。
6. 已知一个顺序表的元素按元素值非递减有序排列，编写一个函数删除向量中多余的值相同的元素。
7. void RSh(int A[n],int k)//把数组A的元素循环右移k位,只用一个辅助存储空间。例如：n=15,k=6
8. 识别读入的一个字符序列是否为反对称的字符序列(以字符′ & ′为对称轴)。
   * 例如:abcd&dcba@ 是反对称字符序列;
   * abc&@ 或 abc&abc@ 或 ab&bac@都不是反对称字符序列。

若没有对称轴怎么办?

1. 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时缀，则可共享相同的后缀存储空间，例如，“loading”和“being”，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）说明你所设计算法的时间和空间复杂度。

1. 有两个栈 s1 和 s2 共享存储空间c(1，m)，其中一个栈底设在c[1]处，另一个栈底设在c[m]处. 假设top1 是栈 1 的栈指针，top2 是栈 2 的栈指针,则两个栈出现上溢和下溢的条件各是什么?
2. 设栈的输入序列是1、2、3、4，则可能是其出栈序列有哪些？
3. 用静态链表实现的队列如右图所示:front = 9, rear = 7. 队列有头结点。当执行下列操作时,队列有什么变化?

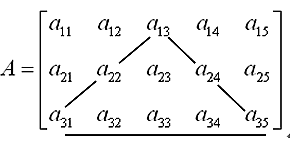
1) 队首元素出队

2) 元素S入队

1. 设有一个矩阵A[1..n, 1..2n-1]，将三个顶点分别为a[1, n]、a[n, 1]和a[n, 2n-1]的三角形区域内所有元素按行存放到一个一维数组B中。下图是n=3的存储示意图，问：

(1) 一维数组*B*要有多少个元素？

(2) 若在一维数组*B*中从0号位置开始存放，则矩阵A中的任一元素*aij*应存于一维数组B的什么下标位置？给出计算公式。



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| B[k] | a13 | a22 | a23 | a24 | a31 | a32 | a33 | a34 | a35 |

1. 若两个稀疏矩阵采用三元组表示，请写出两个矩阵相加的算法

void TSMatrix\_Add(TSMatrix A,TSMatrix B,TSMatrix &C)

//三元组表示的稀疏矩阵加法

1. 利用广义表的HEAD和TAIL操作，写出如上题的表达式，把原子banana从下列广义表中分离出来。

1）L1=(apple,(pear),((banana)),(((orange))))

2）L2=(apple,(pear,(banana),orange));

1. 试编写递归算法，判断两个广义表是否相等。