数组和广义表

▼ 数组顺序存储

- **数组一旦被定义，其维数和每一维的上下界就不再改变**
- **数组的基本操作除初始化和销毁外，只有存取元素和修改元素的值（没有插入和删除）**
- 数组中任一元素不依赖于其他元素，数组是个随机存取结构

两种顺序映象方式

行序为主序

二维R[m][n]：$LOC(i,j)=LOC(0,0)+(i\times n+j)\times L$

三维R[p][m][n]：$LOC(i,j,k)=LOC(0,0,0)+(i\times m\times n+j\times n+k)\times L$

n维数组映象函数：

$R[b_1][b_2]...[b_n]$

$LOC(j_1,j_2,...,j_n)=LOC(0,0,...,0)+\sum_{i=1}^{n}c_ij_i$

其中 $c_n=L,c_{i-1}=b_i\times c_i,1<i\le n$

列序为主序

二维R[m][n]：$LOC(i,j)=LOC(0,0)+(j\times m+i)\times L$

三维R[p][m][n]：$LOC(i,j,k)=LOC(0,0,0)+(k\times p\times m+j\times p+i)\times L$

n维数组映象函数：

$R[b_1][b_2]...[b_n]$

$LOC(j_1,j_2,...,j_n)=LOC(0,0,...,0)+\sum_{i=1}^{n}c_ij_i$

其中 $c_1=L,c_{i+1}=c_i\times b_i,0<i\le n-1$

▼ 题

1. 已知二维数组$A_{\underset{行}{6}\times\underset{列}{8}}$，每个元素占用 6 个字节存储空间，$A$的起始存储位置为 1000，计算：

   a) 数组A占用的存储空间。

   b) 按行优先存储时，$a_{1,4}$的存储地址。(下标从 0 起始)

   c) 按列优先存储时，$a_{3,2}$的存储地址。(下标从 0 起始)

a) $6\times8\times6=288$ 字节

b) $1000+(1\times8+4)\times6=1072$

c) $1000+(2\times6+3)\times6=1090$

2. 已知四维数组$A_{9\times3\times5\times8}$，每个元素占用 4 个字节存储空间，$A$的起始存储位置为 100，请问：

   a) 按行优先时，元素$a_{0,0,0,0}$，$a_{1,1,1,1}$，$a_{3,1,2,5}$，$a_{8,2,4,7}$的存储地址为多少？(下标从 0 起始)

   b) 按列优先排序时，排在第 100 到 104 的 5 个数组元素是什么？(表元素序号以 1 为起始，第 1 个元素$a_{0,0,0,0}$，第 2 个元素为$a_{1,0,0,0}$，依此类推)

a) $a_{0,0,0,0}$: 100

$a_{1,1,1,1}$: $100+(3\times5\times8+5\times8+8+1)\times4=776$

$a_{3,1,2,5}$: $100+(3\times3\times5\times8+5\times8+2\times8+5)\times4=1784$

$a_{8,2,4,7}$: $100+(8\times3\times5\times8+2\times5\times8+4\times8+7)\times4=4416$

b) $99/(9\times3\times5)=0$ 第100个为 $a_{0,2,3,0}$

$99/(9\times3)=3$ 第101个为 $a_{1,2,3,0}$

$(99-9\times3\times3)/9=2$ 第102个为 $a_{2,2,3,0}$

$99-9\times3\times3-9\times2=0$ 第103个为 $a_{3,2,3,0}$

第104个为 $a_{4,2,3,0}$

▼ 特殊矩阵

▼ 对称矩阵

存下三角

矩阵从1开始 数组从0开始

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

a_{ij} 前有 $i-1$ 行，$1+2+\dots+i-1=\frac{i(i-1)}{2}$

a_{ij} 第 i 行前有 $j-1$ 个元素

$a_K=a_{ij}$　$k=\frac{i(i-1)}{2}+j-1$　$(i\geq j)$

$$k=\begin{cases}\frac{i(i-1)}{2}+j-1, & i\geq j\\ \frac{j(j-1)}{2}+i-1, & i<j\end{cases}$$

	a_{11}	a_{21}	a_{22}	$\cdots$	a_{nn}
K	0	1	2	$\cdots$	$\frac{n(n+1)}{2}-1$

矩阵从0开始 数组从0开始

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0n} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1n} \\ a_{20} & a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n-1\,0} & a_{n-1\,1} & a_{n-1\,2} & \cdots & a_{n-1\,n-1} \end{bmatrix}$$

a_{ij} 前有 i 行，$1+2+\dots+i=\frac{i(i+1)}{2}$

a_{ij} 第 i 行前有 j 个元素

$a_K=a_{ij}$　$k=\frac{i(i+1)}{2}+j$　$(i\geq j)$

$$k=\begin{cases}\frac{i(i+1)}{2}+j, & i\geq j\\ \frac{j(j+1)}{2}+i, & i<j\end{cases}$$

	a_{00}	a_{10}	a_{11}	$\cdots$	$a_{n-1\,n-1}$
K	0	1	2	$\cdots$	$\frac{n(n+1)}{2}-1$

矩阵从0开始 数组从1开始

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0n} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1n} \\ a_{20} & a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n-1\,0} & a_{n-1\,1} & a_{n-1\,2} & \cdots & a_{n-1\,n-1} \end{bmatrix}$$

a_{ij} 前有 i 行，$1+2+\dots+i=\frac{i(i+1)}{2}$

a_{ij} 第 i 行前有 j 个元素

$a_K=a_{ij}$　$k=\frac{i(i+1)}{2}+j+1$　$(i\geq j)$

$$k=\begin{cases}\frac{i(i+1)}{2}+j+1, & i\geq j\\ \frac{j(j+1)}{2}+i+1, & i<j\end{cases}$$

	a_{00}	a_{10}	a_{11}	$\cdots$	$a_{n-1\,n-1}$
K	1	2	3	$\cdots$	$\frac{n(n+1)}{2}$

矩阵从1开始 数组从1开始

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

a_{ij} 前有 $i-1$ 行，$1+2+\dots+i-1=\frac{i(i-1)}{2}$

a_{ij} 第 i 行前有 $j-1$ 个元素

$a_K=a_{ij}$　$k=\frac{i(i-1)}{2}+j$　$(i\geq j)$

$$k=\begin{cases}\frac{i(i-1)}{2}+j, & i\geq j\\ \frac{j(j-1)}{2}+i, & i<j\end{cases}$$

	a_{11}	a_{21}	a_{22}	$\cdots$	a_{nn}
K	1	2	3	$\cdots$	$\frac{n(n+1)}{2}$

存上三角

行优先

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

a_{ij} 前有 $i-1$ 行，

$n+(n-1)+\dots+(n-(i-1)+1)=\frac{(2n-i+2)(i-1)}{2}$

a_{ij} 第 i 行前有 $j-i$ 个元素，

$a_K=a_{ij}$　$k=\frac{(2n-i+2)(i-1)}{2}+j-i$

$$k=\begin{cases}\frac{(2n-i+2)(i-1)}{2}+j-i, & i\leq j\\ \frac{(2n-j+2)(j-1)}{2}+i-j, & i>j\end{cases}$$

	a_{11}	a_{12}	a_{13}	$\cdots$	a_{nn}
K	0	1	2	$\cdots$	$\frac{n(n+1)}{2}-1$

列优先

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

a_{ij} 前有 $j-1$ 列，$1+2+\dots+j-1=\frac{j(j-1)}{2}$

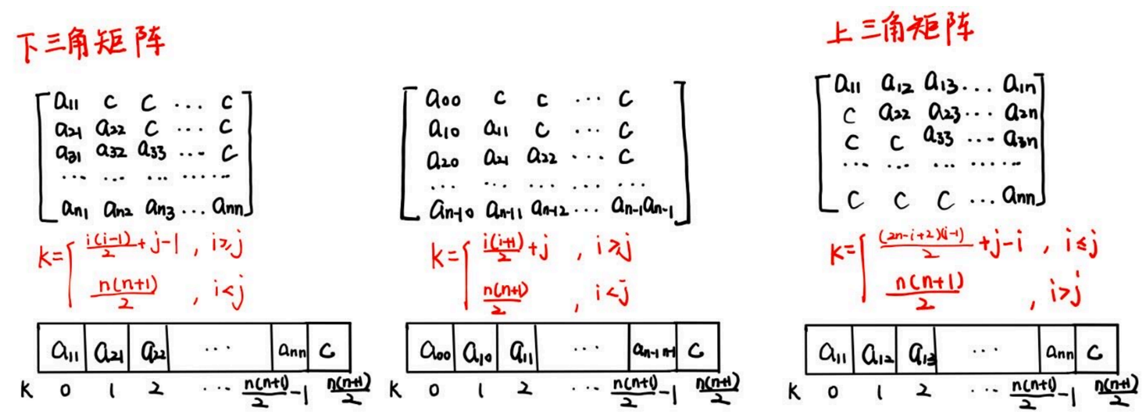
a_{ij} 第 j 列前有 $i-1$ 个元素

$a_K=a_{ij}$　$k=\frac{j(j-1)}{2}+i-1$　$(i\leq j)$

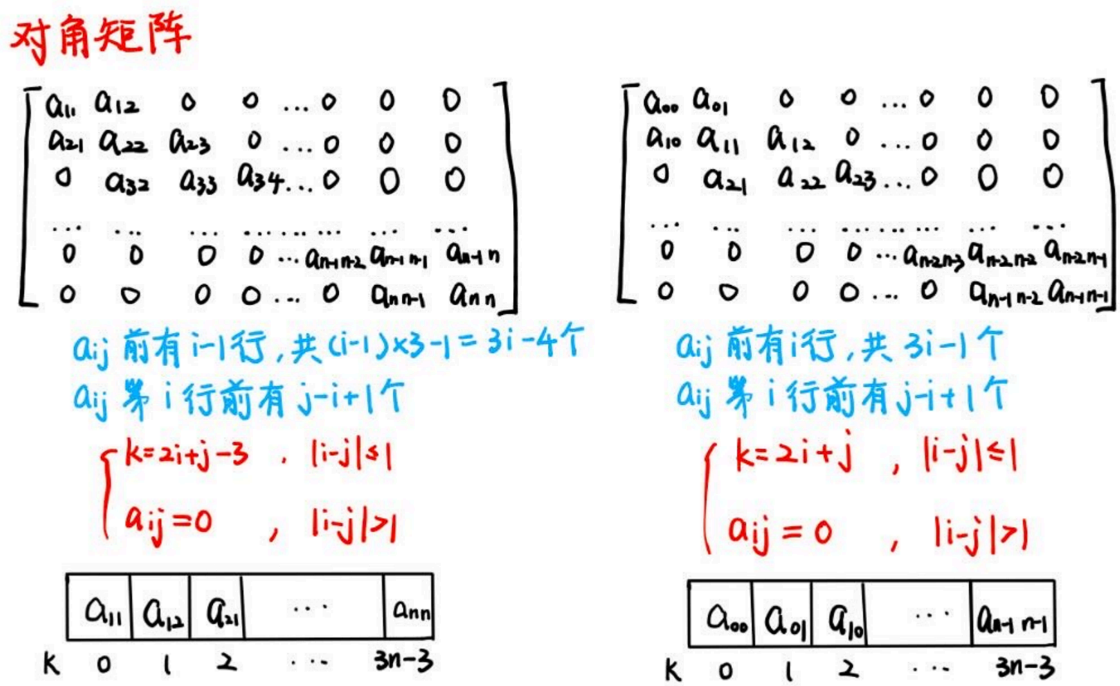
$$k=\begin{cases}\frac{j(j-1)}{2}+i-1, & i\leq j\\ \frac{i(i-1)}{2}+j-1, & i>j\end{cases}$$

	a_{11}	a_{12}	a_{22}	$\cdots$	a_{nn}
K	0	1	2	$\cdots$	$\frac{n(n+1)}{2}-1$

▼ 三角矩阵


下三角矩阵
k= i(i-1)/2+j-1 , i≥j
n(n+1)/2 , i<j
上三角矩阵
k= (2n-i+2)(i-1)/2+j-i , i≤j
n(n+1)/2 , i>j

▼ 对角矩阵


对角矩阵
aij 前有 i-1 行，共 (i-1)×3-1 = 3i-4 个
aij 第 i 行前有 j-i+1 个
k=2i+j-3 , |i-j|≤1
aij=0 , |i-j|>1
aij 前有 i 行，共 3i-1 个
aij 第 i 行前有 j-i+1 个
k=2i+j , |i-j|≤1
aij = 0 , |i-j|>1

▼ 题

设三对角矩阵$A_{n\times n}$所有元素$a_{i,j}$按行优先顺序存储于一维数组$B[3n-2]$中，已知$B[k]=a_{i,j}$，求：

$$\begin{array}{llll} i=0 & \overset{0}{a_{0,0}} & \overset{1}{a_{0,1}} & \\ i=1 & \overset{2}{a_{1,0}} & \overset{3}{a_{1,1}} & \overset{4}{a_{1,2}} \\ i=2 & \overset{5}{a_{2,1}} & \overset{6}{a_{2,2}} & \overset{7}{a_{2,3}} \end{array}$$

a) 用i,j表示k的下标变换公式；（下标i,j,k从 0 起始）

b) 用k表示i,j的下标变换公式；（下标i,j,k从 0 起始）

a) $k = i\times 3-1+j-i+1 = 2i+j$

b) $\begin{cases} i=\lfloor \frac{k+1}{3} \rfloor \\ j=k-2i \end{cases}$

稀疏矩阵

三元组顺序表

$$M=\begin{bmatrix} 0 & 0 & 9 & 0 & -7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 16 & 0 \end{bmatrix}$$

三元组线性表为 ((1,3,9),(1,5,-7),(3,4,8),(4,1,5),(4,6,2),(5,5,16))。

```
typedef struct{
    int i,j;  //行列号
    int e;  //值
}Triple;  //三元组
typedef struct{
    Triple data[MAXSIZE+1];
    int m,n,t;  //行数,列数,非零元素个数
}TSMatrix;
```

转置

压缩转置

- 时间复杂度 $O(n \times t)$ （列数和非零元个数）

1	1	3	9
2	1	5	-7
3	3	4	8
4	4	1	5
5	4	6	2
6	5	5	16

1	1	4	5
2	3	1	9
3	4	3	8
4	5	1	-7
5	5	5	16
6	6	4	2

1.扫描M.data，找出列号为1的元素转置，得到(1,4,5)；
2.扫描M.data，找出列号为2的元素转置，没有则继续；
3.扫描M.data，找出列号为3的元素转置，得到(3,19)。依次类推，则生成的三元组线性表序列为：(1,4,5) (3,1,9) (4,3,8) (5,1,-7) (5,5,16) (6,4,2)

```
bool TransposeMatrix(TSMatrix M,TSMatrix &T){
    T.m=M.n;
    T.n=M.m;
    T.t=M.t;
    if(T.t){
        int q=1;
        //按列枚举
        for(int col=1;col<=M.n;++col){
            //枚举所有三元组
            for(int p=1;p<=M.t;++p){
                if(M.data[p].j==col){  //列标=col
                    T.data[q].i=M.data[p].j;
                    T.data[q].j=M.data[p].i;
                    T.data[q].e=M.data[p].e;
                    ++q;
                }
            }
        }
    }
    return true;
}
```

快速转置

辅助数组：

num存放每列元素的个数

cpot存放每列第一个元素的位置

遍历压缩矩阵表，为每个元素寻找存放位置，即 $cpot[col]$，每次存完 $cpot[col]+1$

- 时间复杂度 $O(n+t)$

```
bool FastTransposeSMatrix(TSMatrix M,TSMatrix &T,int *num,int *cpot)
    T.m=M.n;
    T.n=M.m;
    T.t=M.t;
    if(T.t){
        int col;
        //预处理
        //num存放每列元素个数
        for(col=1;col<=M.n;++col){
            num[col]=0;
        }
        for(int t=1;t<=M.t;++t){
            num[M.data[t].j]++;
        }
        //cpot存放每列第一个元素在M中的位置
        cpot[1]=1;
        for(col=2;col<=M.n;++col){
            cpot[col]=cpot[col-1]+num[col-1];
        }
        //转置
        int q;
        for(int p=1;p<=M.t;++p){
            col=M.data[p].j;
            q=cpot[col]; //找到该存放的位置
            T.data[q].i=M.data[p].j;
```

```
            T.data[q].j=M.data[p].i;
            T.data[q].e=M.data[p].e;
            cpot[col]++;
        }
        return true;
    }
    return false;
}
```

▼ 行逻辑链接顺序表

增加一个rpos数组存放每行第一个元素的下标

```
typedef struct{
    int i,j;
    int e;
}Triple;
typedef struct{
    Triple data[MAXSIZE+1];
    int rpos[MAXRC+1];  //各行第一个非零元在data中位置
    int m,n,t;
}RLSMatrix;
```

矩阵乘法

辅助数组：

$ctemp$ 存放 Q 的第 i 行每列的计算结果

按行计算，遍历 M 在第 i 行的元素 p_k，找出 N 中行号等于 p_k 列号的元素，$M(i,k) \times N(k,j)$ 结果存放在 $ctemp[j]$。再存整行$Q(i,j) = ctemp[j]$

M	i	j	e
1	1	2	1
2	2	1	2
3	2	3	-4
4	3	1	-1
5	4	2	-2
6	4	3	3

M.rops

1	2	3	4	(5)
1	2	4	5	7

N	i	j	e
1	1	1	2
2	1	4	6
3	2	2	-4
4	3	2	-1
5	3	4	3

N.rops

1	2	3	(4)
1	3	4	6

Q	i	j	e
1	1	2	-4
2	2	1	4
3	2	2	4
4	3	1	-2
5	3	4	-6
6	4	2	5
7	4	4	9

Q.rops

1	2	3	4	(5)
1	2	4	6	8

第1行 ctemp[2] += M(1,2) × N(2,2) = -4

Q(1,2) = -4

第2行 ctemp[1] += M(2,1) × N(1,1) = 4

ctemp[4] += M(2,1) × N(1,4) = 12

ctemp[2] += M(2,3) × N(3,2) = 4

ctemp[4] += M(2,3) × N(3,4) = -12

Q(2,1) = 4 Q(2,2) = 4 Q(2,4) = 0

第3行 ctemp[1] += M(3,1) × N(1,1) = -2

ctemp[4] += M(3,1) × N(1,4) = -6

Q(3,1) = -2 Q(3,4) = -6

第4行 ctemp[2] += M(4,2) × N(2,2) = 8

ctemp[2] += M(4,3) × N(3,2) = -3

ctemp[4] += M(4,3) × N(3,4) = 9

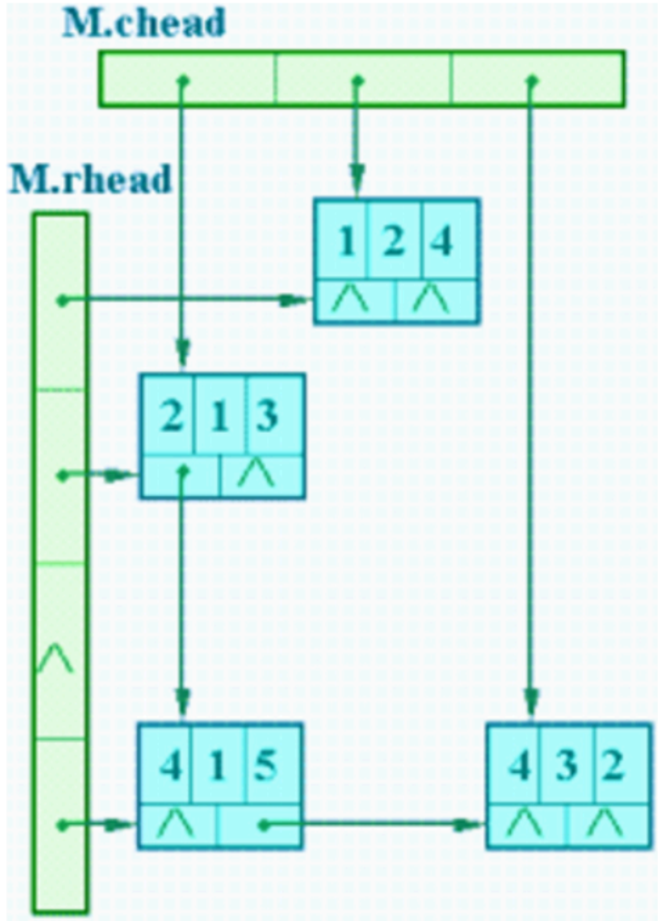
Q(4,2) = 5 Q(4,4) = 9

```
bool MulSMatrix(RLSMatrix M,RLSMatrix N,RLSMatrix &Q){
    if(M.n!=N.m) return false;
    Q.m=M.m;
```

```
    Q.n=N.n;
    Q.t=0;
    //M,N均有非零元
    if(M.t*N.t!=0){
        //求Q的一行
        for(int arow=1;arow<=M.m;arow++){
            int ctemp[Q.n+1]={0}; //累加器清零
            Q.rpos[arow]=Q.t+1; //当前行在Q中起始位置
            for(int p=M.rpos[arow];p<M.rpos[arow+1];p++){
                //遍历M在arow行的所有值
                int brow=M.data[p].j;
                //取出该元素所在M的列号,对应N的行号
                for(int q=N.rpos[brow];q<N.rpos[brow+1];q++){
                    int ccol=N.data[q].j;  //乘积元素在Q中的列号
                    ctemp[ccol]+=M.data[p].e*N.data[q].e;
                }
            }
            //存储该行非零元
            for(int ccol=1;ccol<=Q.n;++ccol){
                if(ctemp[ccol]){
                    ++Q.t;
                    if(Q.t>MAXSIZE) return false;
                    Q.data[Q.t].i=arow;
                    Q.data[Q.t].j=ccol;
                    Q.data[Q.t].e=ctemp[ccol];
                }
            }
            //添加一个行数+1的rpos指向末尾+1
            //方便计算最后一行个数=rpos[k+1]-rpos[k]
            Q.rpos[M.m+1]=Q.t+1;
        }
    }
    return true;
}
```

▼ 十字链表

$$M = \begin{bmatrix} 0 & 4 & 0 \\ 3 & 0 & 0 \\ 0 & 0 & 0 \\ 5 & 0 & 2 \end{bmatrix}$$




```
//结点
typedef struct OLNode{
    int i,j;
    int e;
    struct OLNode *right,*down;
}OLNode,*OLink;

//链表
typedef struct{
    OLink *rhead,*chead; //rhead行表头,chead列表头,指向OLink
    int m,n,t;
}CrossList;
```

▼ 广义表

- **长度：**最外层（）的元素个数
- **深度：**（）最大重数
- **表头：**第一个元素
- **表尾：**其余元素构成的广义表

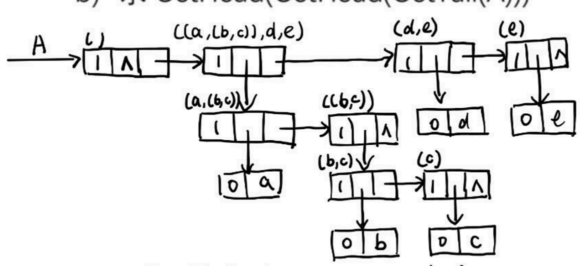
```
typedef enum{
    ATOM=0;    //标记原子
    LIST=1;       //标记子表
}ElemTag;
typedef struct GLNode{
    ElemTag tag;   //区分原子结点和表结点
    union{
        AtomType data;  //原子结点值域
        struct{
            struct GLNode *hp,*tp;
        }ptr; //ptr表结点指针域
    };
}*GList;
```

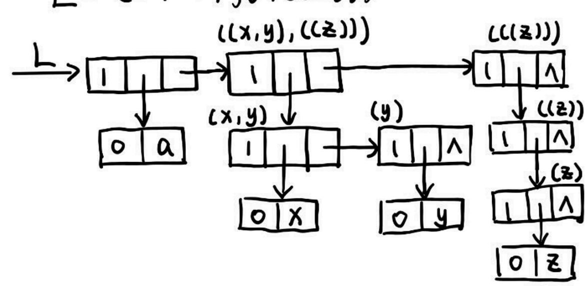
4, 已知广义表 A=((), (a, (b, c)), d, e) ,

a) 画出 A 的链式存储结构

b) 求 GetHead(GetHead(GetTail(A)))

L = (a, (x, y), ((z)))





GetTail(A) = ((a,(b,c)),d,e)

GetHead(GetTail(A)) = (a,(b,c))

GetHead(GetHead(GetTail(A)))=a