# 第五章数组和广义表

# 数组和广义表

1. 数组的定义
2. 数组的顺序表示和实现
3. 矩阵的压缩存储
4. 广义表的定义
5. 广义表的存储结构

# 数组的定义

- $k$维数组D＝{$a_{j1,j2,...,jk}$|$k$>0称为数组的维数, $b_i$是数组第$i$维的长度，$j_i$是数组元素第$i$维的下标，$a_{j1,j2,...,jk}$属于ElemSet}
- $j_i=0, ..., b_i-1,\ i=1, 2, …, k$

说明：$a_{j1,j2,...,jn}$属于ElemSet表示同一数组的数据元素性质相同

# 二维数组: 假设$b_1=m,b_2=n$

$$A_{m\times n}=\begin{bmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0,n-1} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m-1,0} & a_{m-1,1} & a_{m-1,2} & & a_{m-1,n-1} \end{bmatrix}$$

看成n个数据元素的线性表，每个数据元素代表数组中的一列

$A_{m\times n}=(R_0,R_1,\ldots,R_{n-1})$

$R_0=(a_{00},a_{10},\ldots,a_{m-1,0})$

$R_i=(a_{0,i},a_{1,i},\ldots,a_{m-1,i})$

看成m个数据元素的线性表，每个数据元素代表数组中的一行

$A_{m\times n}=(S_0,S_1,\ldots,S_{m-1})$

$S_0=(a_{00},a_{01},\ldots,a_{0,n-1})$

$S_i=(a_{i,0},a_{i,1},\ldots,a_{i,n-1})$

# 数组的基本操作

- 初始化一个数组
- 取数组元素的值
- 给数组元素赋值

说明：

- 数组一般不做插入、删除操作----多采用顺序存储结构
- 数组是多维的结构，而存储空间是一个一维的结构，数组顺序存储就是把多维结构存为一维结构

# 数组的顺序表示和实现

- 二维数组有两种顺序映象的方式:

1. 以行序为主序
2. 以列序为主序

- 多维数组同样有两种顺序映象的方式

二维数组的存储方法

**特点：有地址计算公式，可以随机访问**

# 以“行序为主序”的存储映象

| | | |
|---|---|---|
| $a_{0,0}$ | $a_{0,1}$ | $a_{0,2}$ |
| $a_{1,0}$ | $a_{1,1}$ | $a_{1,2}$ |

2行3列的二维数组

→

| | | | | | |
|---|---|---|---|---|---|
| $a_{0,0}$ | $a_{0,1}$ | $a_{0,2}$ | $a_{1,0}$ | $a_{1,1}$ | $a_{1,2}$ |

L

以“行序为主序”存于一维的存储空间

二维数组以“行序为主序”存到一维存储空间—存到一维数组

二维数组A中任一元素$a_{i,j}$的存储位置：

$$LOC(a_{i,j}) = LOC(a_{0,0}) + (b_2 \times i + j) \times L$$

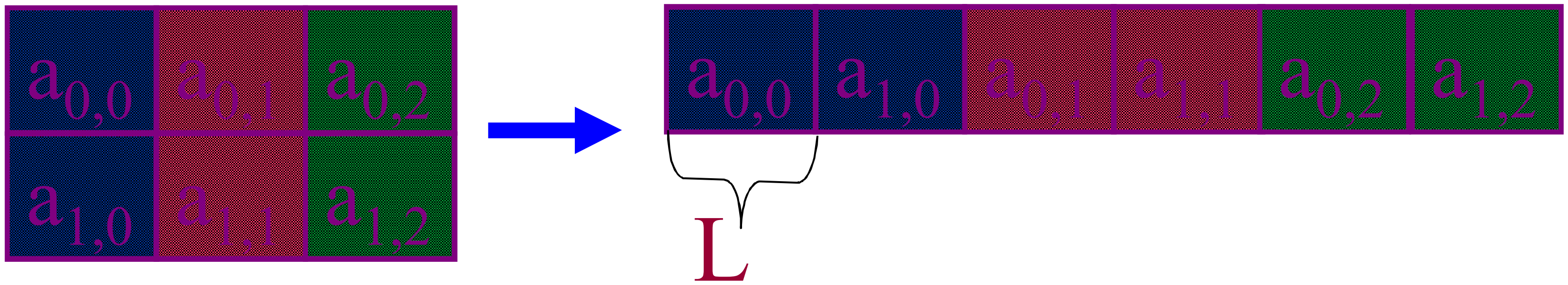
称为基地址或基址

- 从数组的第一行开始依次存放每一行的数组元素；
- 存放第i行时，从第一列开始顺次存放

二维数组的存储方法

**特点：有地址计算公式，可以随机访问**

# 以“列序为主序”的存储映象




2行3列的二维数组

二维数组以“列序为主序”存到一维存储空间—存到一维数组

二维数组A中任一元素$a_{i,j}$ 的存储位置：

$$LOC(a_{i,j}) = LOC(a_{0,0}) + (b_1 \times j + i) \times L$$

称为基地址或基址

- 从数组的第一列开始依次存放每一列的数组元素；
- 存放第i列时，从第一行开始顺次存放

# 矩阵（二维数组）的压缩存储

宗旨：为值相同的矩阵元素只分配一个空间，对零元不分配存储空间

- 研究二类矩阵的压缩存储：

•特殊矩阵：非零元在矩阵中的分布有一定规则

1. 上（下）三角矩阵
2. 对称矩阵
3. 对角矩阵

•稀疏阵：零元多，分布无规律

- 设计的压缩存储方式要方便访问操作，最好仍能"随机访问"

# 1、上三角矩阵 $(a_{ij})_{n\times n}$, $1\le i, j \le n$

特点：$i>j$时，$a_{ij}=0$ 或常量$C$

$$
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\
0 & a_{22} & a_{23} & \dots & a_{2n} \\
0 & 0 & a_{33} & \dots & a_{3n} \\
 & . & & & . \\
 & . & & & . \\
 & . & & & . \\
0 & 0 & 0 & \dots & a_{nn}
\end{bmatrix}
\qquad
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\
c & a_{22} & a_{23} & \dots & a_{2n} \\
c & c & a_{33} & \dots & a_{3n} \\
 & . & & & . \\
 & . & & & . \\
 & . & & & . \\
c & c & c & \dots & a_{nn}
\end{bmatrix}
$$

上三角矩阵示例1　　　　上三角矩阵示例2

存储方式：列为主序压缩存储和行为主序压缩存储，存储空间是一维的，将二维数组以一维方式存储

特点：均可以随机访问数组元素

# 1、上三角矩阵—列为主序压缩存储 –数组sa[M]

特点：$i>j$时，$a_{ij}=0$ 或常量$C$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ & . & & & . \\ & . & & & . \\ & . & & & . \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

$i \le j$时，$a_{ij}$为非0元，存放地址Loc($a_{ij}$)的计算公式：

$$Loc(a_{ij}) = Loc(a_{11}) + ((j-1)j/2 + i - 1) * L$$

一维存储空间用一维数组sa[M]表示，Loc($a_{ij}$)计算公式（$a_{11}$存于sa[0],地址为0）：

$$Loc(a_{ij}) = 0 \qquad +((j-1)j/2 + i - 1) * 1$$

$Loc(a_{ij}) = k = (j-1)j/2 + i - 1$，$a_{ij} (i \le j)$存于下标为$k$的数组元素中。

数组的大小M=? M=n(n+1)/2

一维存储空间

| sa[0] | sa[1] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $a_{11}$ | $a_{12}$ | $a_{22}$ | $a_{13}$ | $a_{23}$ | $\cdots$ | $a_{1n}$ | … | $a_{nn}$ |

特点：有地址计算公式，可以随机访问

列为主序压缩存储：从第一列开始依次存放每一列的“非0元”

一维存储空间—一维数组

# 1、上三角矩阵——列为主序压缩存储 –数组sa[M]

特点：$i>j$时，$a_{ij}$= 常量C

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ c & a_{22} & a_{23} & \cdots & a_{2n} \\ c & c & a_{33} & \cdots & a_{3n} \\ & \cdot & & & \cdot \\ & \cdot & & & \cdot \\ & \cdot & & & \cdot \\ c & c & c & \cdots & a_{nn} \end{bmatrix}$$

数组sa的大小M=？

M=n(n+1)/2+1

i≤j时，$a_{ij}$为非C元，存放地址Loc($a_{ij}$)计算公式：

Loc($a_{ij}$)= ($j$-1)$j$/2+$i$-1

常量C的存放地址：

n(n+1)/2

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $a_{11}$ | $a_{12}$ | $a_{22}$ | $a_{13}$ | $a_{23}$ | $\cdots$ | $a_{1,n}$ | $\cdots$ | $a_{n,n}$ | $C$ |

sa[0] sa[1] sa[n(n+1)/2-1]

列为主序压缩存储：从第一列开始依次存放每一列的“非C元”

最后一个数组元素存常量$C$

## 2、下三角矩阵$(a_{ij})_{n\times n}$, $1\le i,j \le n$

特点：$i<j$时，$a_{ij}=0$ 或常量$C$

$$A_{n\times n}=\begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & & a_{nn} \end{bmatrix}$$

下三角矩阵示例1

$$A_{n\times n}=\begin{bmatrix} a_{11} & c & c & \cdots & c \\ a_{21} & a_{22} & c & \cdots & c \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & & a_{nn} \end{bmatrix}$$

下三角矩阵示例2

存储方式：列为主序压缩存储和行为主序压缩存储，存储空间是一维的，将二维数组以一维方式存储

特点：均可以随机访问数组元素

行为主序压缩存储：从第一行开始依次存放每一行的“非0（C）元”

# 2、下三角矩阵—行为主序压缩存储 –数组sa[M]

特点：$i<j$时，$a_{ij}=0$ 或常量$C$

$$A_{n\times n}=\begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & & a_{nn} \end{bmatrix}$$

$i \geq j$时，$a_{ij}$为非0元，存放地址Loc($a_{ij}$)的计算公式：

$$Loc(a_{ij}) = Loc(a_{11}) + ((i-1)i/2 + j-1) * L$$

一维存储空间用一维数组sa[M]表示，Loc($a_{ij}$)计算公式（$a_{11}$存于sa[0],地址为0）：

$$Loc(a_{ij}) = 0 \qquad +((i-1)i/2+j-1)*1$$

$Loc(a_{ij})=k=(i-1)i/2+j-1$，$a_{ij}(i \geq j)$存于下标为$k$的数组元素中。

数组的大小M=? M=n(n+1)/2

一维存储空间

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $a_{11}$ | $a_{21}$ | $a_{22}$ | $a_{31}$ | $a_{32}$ | $\cdots$ | $a_{n1}$ | … | $a_{nn}$ |

行为主序压缩存储：从第一行开始依次存放每一行的“非0（C）元”

一维存储空间——一维数组

特点：有地址计算公式，可以随机访问

# 2、下三角矩阵—行为主序压缩存储 –数组sa[M]

特点：$i<j$时，$a_{ij}=0$ 或常量$C$

**数组sa的大小M=？**

M=n(n+1)/2+1

$$A_{n\times n}=\begin{bmatrix} a_{11} & c & c & \cdots & c \\ a_{21} & a_{22} & c & \cdots & c \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & & a_{nn} \end{bmatrix}$$

下三角矩阵示例2

i≥j时，$a_{ij}$为非C元，存放地址Loc($a_{ij}$)计算公式：

$$\mathrm{Loc}(a_{ij}) = (i-1)i/2+j-1$$

常量C的存放地址：

n(n+1)/2

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $a_{11}$ | $a_{21}$ | $a_{22}$ | $a_{31}$ | $a_{32}$ | $\cdots$ | $a_{n1}$ | … | $a_{nn}$ | $C$ |

行为主序压缩存储：从第一行开始依次存放每一行的“非C元”

## 3、对称矩阵，特点：$a_{ij} = a_{ji}$

存放方式：只存上三角阵或只存下三角阵都可以

# 4、对角矩阵 -- 2d+1对角阵：主对角线和主对角线上面d条对角线、主对角线下面d条对角线上的数据元素分布不规律，非0（C）

$$\begin{bmatrix} a_{00} & a_{01} & 0 & 0 & 0 & \cdots & 0 \\ a_{10} & a_{11} & a_{12} & 0 & 0 & \cdots & 0 \\ 0 & a_{21} & a_{22} & a_{23} & 0 & \cdots & 0 \\ 0 & 0 & a_{32} & a_{33} & a_{34} & \cdots & 0 \\ 0 & 0 & 0 & a_{43} & a_{44} & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & \cdots & a_{n-1,n-1} \end{bmatrix}$$

3-对角阵

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & 0 & 0 & \cdots & 0 \\ a_{10} & a_{11} & a_{12} & a_{13} & 0 & \cdots & 0 \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & \cdots & 0 \\ 0 & a_{31} & a_{32} & a_{33} & a_{34} & \cdots & 0 \\ 0 & 0 & a_{42} & a_{43} & a_{44} & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & \cdots & a_{n-1,n-1} \end{bmatrix}$$

5-对角阵

2d+1 对角阵**特点**：第一行和最后一行每行有d+1个数据元素，余下每行最多2d+1个数据元素

**压缩存储方法：第一行和最后一行各存d+1个数据元素，余下每行存2d+1个数据元素**

压缩存储方法：第一行和最后一行各存d+1个数据元素，余下每行存2d+1个数据元素

$$
\begin{bmatrix}
a_{00} & a_{01} & 0 & 0 & 0 & \cdots & 0 \\
a_{10} & a_{11} & a_{12} & 0 & 0 & \cdots & 0 \\
0 & a_{21} & a_{22} & a_{23} & 0 & \cdots & 0 \\
0 & 0 & a_{32} & a_{33} & a_{34} & \cdots & 0 \\
0 & 0 & 0 & a_{43} & a_{44} & \cdots & 0 \\
. & . & . & . & . & . & . \\
0 & 0 & 0 & 0 & 0 & \cdots & a_{n-1.n-1}
\end{bmatrix}
$$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_{00}$ | $a_{01}$ | $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{21}$ | $a_{22}$ | $a_{23}$ | | | | | | | | |

3-对角阵行为主序压缩存储

压缩存储方法：第一行和最后一行各存d+1个数据元素，余下每行存2d+1个数据元素

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & 0 & 0 & \cdots & 0 \\ a_{10} & a_{11} & a_{12} & a_{13} & 0 & \cdots & 0 \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & \cdots & 0 \\ 0 & a_{31} & a_{32} & a_{33} & a_{34} & \cdots & 0 \\ 0 & 0 & a_{42} & a_{43} & a_{44} & \cdots & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & \cdots & a_{n-1.n-1} \end{bmatrix}$$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_{00}$ | $a_{01}$ | $a_{02}$ | | $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ | $a_{20}$ | $a_{21}$ | $a_{22}$ | $a_{23}$ | $a_{23}$ | | | |

5-对角阵行为主序压缩存储

## 2d+1-对角阵行为主序压缩存储地址计算公式

- 矩阵元素下表从0开始的地址计算公式:
  - $Loc(a_{ij})=Loc(a_{00})+(2d+1)*i-d+j-(i-d)$
  - $0 \le i,j \le n-1,\ |i-j| \le d$
- 矩阵元素下表从1开始的地址计算公式:
  - $Loc(a_{ij})=Loc(a_{11})+(2d+1)*(i-1)-d+j-i+d$

    $= Loc(a_{11})+(2d+1)*(i-1)+j-I$
  - $1 \le i,j \le n,\ |i-j| \le d$

## 4. 稀疏矩阵:零元多，在矩阵中随机出现

假设 m 行 n 列的矩阵含 t 个非零元素，则称

$$\delta = \frac{t}{m \times n}$$

为稀疏因子。

通常认为 $\delta \le 0.05$ 的矩阵为稀疏矩阵。

常规存储方法缺点:

1) 零值元素占了很大空间;

2) 计算中进行了很多和零值的运算，遇除法，还需判别除数是否为零。

## 解决问题的原则：

1) 尽可能少存或不存零值元素；

2) 尽可能减少没有实际意义的运算；

3) 操作方便。即：尽可能快地找到与下标值(i,j)对应的元素，尽可能快地找到同一行或同一列的非零值元。

## 稀疏矩阵的压缩存储方法：

一、三元组顺序表

二、行逻辑联接的顺序表

三、 十字链表

# 一、三元组顺序表

- 采用一维数组以行为主序存放每一非零元;
- 每一非零元只存行号、列号、非零元的值

```
#define  MAXSIZE  12500
typedef struct {
    int  i, j;     //该非零元的行下标和列下标
    ElemType  e;     // 该非零元的值
} Triple;  // 三元组类型
typedef struct {
    Triple  data[MAXSIZE];
    int     mu, nu, tu; //矩阵的行数、列数和非0元总数
} TSMatrix;  // 稀疏矩阵类型
```

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix}$$

稀疏矩阵

| $i$ | $j$ | $e$ | |
|---|---|---|---|
| | | | M.data[0] |
| 1 | 2 | 14 | M.data[1] |
| 1 | 5 | -5 | M.data[2] |
| 2 | 2 | -7 | M.data[3] |
| 3 | 1 | 36 | M.data[4] |
| 3 | 4 | 28 | M.data[5] |

TSMatrix M； //稀疏矩阵对应的三元组顺序表

| | | |
|---|---|---|
| | | |
| 1 | 5 | -5 |
| 1 | 2 | 14 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

| | | |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 3 | 1 | 36 |
| 2 | 2 | -7 |
| 3 | 4 | 28 |

**三元组顺序表要求：**

**非零元以行为主序顺序存放**

# 一、三元组顺序表运算

## ■ 矩阵运算的实现:

- ➢ 矩阵转置
- $A=(a_{ij})_{m\times n}$转置为$B=(b_{ji})_{n\times m}$
- $a_{ij}=b_{ji}$
- B阵仍为稀疏矩阵，采用**三元组顺序表**存放
- ➢ 矩阵相加
- C=A+B要求A和B行数、列数相同
- A和B对应位置的矩阵元素相加
- C采用**三元组顺序表**存放

# 矩阵转置

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \xrightarrow{\text{转置}} \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

矩阵的三元组顺序表

转置

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| 4 | 3 | 28 |
| 5 | 1 | -5 |

转置后矩阵的三元组顺序表

非零元以行为主序顺序存放

# 矩阵转置

方法1：

- 根据三元组顺序表的特点，首先扫描一遍三元组，将扫描到的列号为1的非0元的行列交换存放于转置后的新阵，生成新阵第一行的非0元；
- 再扫描一遍三元组，将扫描到的列号为2的非0元的行列交换存放于转置后的新阵，生成新阵第二行的非0元；
- ……

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| **1** | **2** | **14** |
| **1** | **5** | **-5** |
| **2** | **2** | **-7** |
| **3** | **1** | **36** |
| **3** | **4** | **28** |

B

| *i* | *j* | *e* |
|---|---|---|
| | | |
| **1** | **3** | **36** |
| **2** | **1** | **14** |
| **2** | **2** | **-7** |
| **4** | **3** | **28** |
| **5** | **1** | **-5** |

**方法1:**

- 原阵有nu列，转置后的新阵就有nu行，要生成新阵，就要对原阵的非0元扫描nu遍

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| **1** | **2** | **14** |
| **1** | **5** | **-5** |
| **2** | **2** | **-7** |
| **3** | **1** | **36** |
| **3** | **4** | **28** |

**col=1**

← p=1

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

← q=1

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

col=1

p=1

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

q=1

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

col=1

p=3

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

q=1

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| **1** | **2** | **14** |
| **1** | **5** | **-5** |
| **2** | **2** | **-7** |
| **3** | **1** | **36** |
| **3** | **4** | **28** |

← p=4

**col=1**

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

← q=1

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

col=1

p=4

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| | | |
| | | |
| | | |
| | | |

q=1

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| **1** | **2** | **14** |
| **1** | **5** | **-5** |
| **2** | **2** | **-7** |
| **3** | **1** | **36** |
| **3** | **4** | **28** |

← p=5

col=1

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| **1** | **3** | **36** |
| | | |
| | | |
| | | |
| | | |

← q=2

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=1

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p=6

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| | | |
| | | |
| | | |
| | | |

q=2

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=2

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| **1** | **2** | **14** |
| **1** | **5** | **-5** |
| **2** | **2** | **-7** |
| **3** | **1** | **36** |
| **3** | **4** | **28** |

p=1

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| **1** | **3** | **36** |
| | | |
| | | |
| | | |
| | | |

q=2

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

**p<=A.tu**

**col=2**

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| **1** | **2** | **14** |
| **1** | **5** | **-5** |
| **2** | **2** | **-7** |
| **3** | **1** | **36** |
| **3** | **4** | **28** |

← **p=1**

B

| *i* | *j* | *e* |
|---|---|---|
| | | |
| **1** | **3** | **36** |
| **2** | **1** | **14** |
| | | |
| | | |
| | | |

← **q=2**

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p<=A.tu

col=2

p=2

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| | | |
| | | |
| | | |

q=3

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p<=A.tu

col=2

p=3

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| | | |
| | | |
| | | |

q=3

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=2

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=3

B

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

← q=3

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=2

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=4

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

← q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

A

| i | j | e |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=5

col=2

B

| i | j | e |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

← q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=2

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=6

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

← q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ | |
|---|---|---|---|
| | | | |
| 1 | 2 | 14 | ← p=1 |
| 1 | 5 | -5 | |
| 2 | 2 | -7 | |
| 3 | 1 | 36 | |
| 3 | 4 | 28 | |

p<=A.tu

col=3

B

| $i$ | $j$ | $e$ | |
|---|---|---|---|
| | | | |
| 1 | 3 | 36 | |
| 2 | 1 | 14 | |
| 2 | 2 | -7 | |
| | | | ← q=4 |
| | | | |

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=3

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

$\leftarrow$ p=2

B

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

$\leftarrow$ q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p<=A.tu

col=3

p=3

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=3

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=4

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

← q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=3

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=5

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

← q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p<=A.tu

col=3

p=6

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=4

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p=1

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=4

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p=2

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=4

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=3

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

← q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=4

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=4

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

← q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p<=A.tu

col=4

p=5

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| | | |
| | | |

q=4

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Longrightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p<=A.tu

col=4

p=6

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| 4 | 3 | 28 |
| | | |

q=5

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=5

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=1

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| 4 | 3 | 28 |
| | | |

← q=5

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p<=A.tu

col=5

p=2

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| 4 | 3 | 28 |
| | | |

q=5

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=5

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=3

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| 4 | 3 | 28 |
| 5 | 5 | -1 |

← q=6

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p<=A.tu

col=5

← p=4

B

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| 4 | 3 | 28 |
| 5 | 5 | -1 |

← q=6

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p<=A.tu

col=5

p=5

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| 4 | 3 | 28 |
| 5 | 5 | -1 |

q=6

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

p<=A.tu

col=5

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=6

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| 4 | 3 | 28 |
| 5 | 5 | -1 |

← q=6

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p<=A.tu

col=6

col<=T.nu

p=6

B

| $i$ | $j$ | $e$ |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| 4 | 3 | 28 |
| 5 | 5 | -1 |

q=6

# 如何求转置矩阵?

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

B

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 3 | 36 |
| 2 | 1 | 14 |
| 2 | 2 | -7 |
| 4 | 3 | 28 |
| 5 | 1 | -5 |

方法1：将矩阵A转置成矩阵B

```
Status TransposeSMatrix(TSMatrix A, TSMatrix &B){
  B.mu = A.nu;  B.nu = A.mu;  B.tu = A.tu;
  if (B.tu) {
    q = 1;
    for (col=1; col<=A.nu; ++col)
      for (p=1; p<=A.tu; ++p)
        if(A.data[p].j == col) {
          B.data[q].i = A.data[p].j; B.data[q].j = A.data[p].i;
          B.data[q].e = A.data[p].e; q++;}
  }
  return OK;
} // TransposeSMatrix
```

时间复杂度为：O(A.nu*A.tu)

缺点：时间效率低

## 方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的**当前存放**的位置

初始时为每一行第一个非零元存放的位置

所谓“位置”，即在三元组中存放的数组元素的下标

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← t=1

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 0 | 0 | 0 | 0 | 0 |
| Cpot[col] | | | | | |

## 方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的**当前存放**的位置

初始时为每一行第一个非零元存放的位置

**所谓"位置"，即在三元组中存放的数组元素的下标**

| *i* | *j* | *e* | |
|---|---|---|---|
| | | | |
| 1 | 2 | 14 | ← t=1 |
| 1 | 5 | -5 | |
| 2 | 2 | -7 | |
| 3 | 1 | 36 | |
| 3 | 4 | 28 | |

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 0 | 1 | 0 | 0 | 0 |
| Cpot[col] | | | | | |

方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的**当前存放**的位置

初始时为每一行第一个非零元存放的位置

所谓“位置”，即在三元组中存放的数组元素的下标

| *i* | *j* | *e* | |
|---|---|---|---|
| | | | |
| 1 | 2 | 14 | |
| 1 | 5 | -5 | ← t=2 |
| 2 | 2 | -7 | |
| 3 | 1 | 36 | |
| 3 | 4 | 28 | |

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 0 | 1 | 0 | 0 | 0 |
| Cpot[col] | | | | | |

方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的当前存放的位置

初始时为每一行第一个非零元存放的位置

所谓“位置”，即在三元组中存放的数组元素的下标

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

t=2

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 0 | 1 | 0 | 0 | 1 |
| Cpot[col] | | | | | |

方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的当前存放的位置

初始时为每一行第一个非零元存放的位置

所谓“位置”，即在三元组中存放的数组元素的下标

| i | j | e |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← t=3

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 0 | 1 | 0 | 0 | 1 |
| Cpot[col] | | | | | |

方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的当前存放的位置

初始时为每一行第一个非零元存放的位置

所谓“位置”，即在三元组中存放的数组元素的下标

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← t=4

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 0 | 2 | 0 | 0 | 1 |
| Cpot[col] | | | | | |

方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的**当前存放**的位置

初始时为每一行第一个非零元存放的位置

**所谓“位置”，即在三元组中存放的数组元素的下标**

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← t=5

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 0 | 1 |
| Cpot[col] | | | | | |

方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的当前存放的位置

初始时为每一行第一个非零元存放的位置

所谓“位置”，即在三元组中存放的数组元素的下标

t<=A.tu

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

t=6

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | | | | | |

```
cpot[1] = 1;
for (col=2; col<=A.nu; ++col)
   cpot[col] = cpot[col-1] + num[col-1];
```

## 方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的当前存放的位置

初始时为每一行第一个非零元存放的位置

所谓“位置”，即在三元组中存放的数组元素的下标

t<=A.tu

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← t=6

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | | | | |

```
cpot[1] = 1;
for (col=2; col<=A.nu; ++col)
   cpot[col] = cpot[col-1] + num[col-1];
```

方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的当前存放的位置

初始时为每一行第一个非零元存放的位置

所谓"位置"，即在三元组中存放的数组元素的下标

t<=A.tu

| i | j | e |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← t=6

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | 2 | | | |

```
cpot[1] = 1;
for (col=2; col<=A.nu; ++col)
  cpot[col] = cpot[col-1] + num[col-1];
```

## 方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的当前存放的位置

初始时为每一行第一个非零元存放的位置

所谓"位置"，即在三元组中存放的数组元素的下标

t<=A.tu

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← t=6

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | 2 | 4 | | |

```
cpot[1] = 1;
for (col=2; col<=A.nu; ++col)
  cpot[col] = cpot[col-1] + num[col-1];
```

方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的当前存放的位置

初始时为每一行第一个非零元存放的位置

所谓“位置”，即在三元组中存放的数组元素的下标

t<=A.tu

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← t=6

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | 2 | 4 | 4 | |

```
cpot[1] = 1;
for (col=2; col<=A.nu; ++col)
  cpot[col] = cpot[col-1] + num[col-1];
```

方法2：减少原阵的扫描次数，提高时间效率

Num[col]:存放矩阵B中每一行非零元的个数

Cpot[col]:存放矩阵B中每一行非零元的当前存放的位置

初始时为每一行第一个非零元存放的位置

所谓"位置"，即在三元组中存放的数组元素的下标

t<=A.tu

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← t=6

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | 2 | 4 | 4 | 5 |

```
cpot[1] = 1;
for (col=2; col<=A.nu; ++col)
   cpot[col] = cpot[col-1] + num[col-1];
```

方法2：

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | 2 | 4 | 4 | 5 |

p<=A.tu

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=1

B

| | *i* | *j* | *e* |
|---|---|---|---|
| 0 | | | |
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |

方法2：

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | 2 | 4 | 4 | 5 |

p<=A.tu

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=1

B

| *i* | *j* | *e* | |
|---|---|---|---|
| | | | 0 |
| | | | 1 |
| 2 | 1 | 14 | 2 |
| | | | 3 |
| | | | 4 |
| | | | 5 |

## 方法2：

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | **3** | 4 | 4 | 5 |

**p<=A.tu**

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=2

B

| | *i* | *j* | *e* |
|---|---|---|---|
| 0 | | | |
| 1 | | | |
| 2 | 2 | 1 | 14 |
| 3 | | | |
| 4 | | | |
| 5 | | | |

## 方法2：

p<=A.tu

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | 3 | 4 | 4 | 5 |

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p=2

B

| | *i* | *j* | *e* |
|---|---|---|---|
| 0 | | | |
| 1 | | | |
| 2 | 2 | 1 | 14 |
| 3 | | | |
| 4 | | | |
| 5 | 5 | 1 | -5 |

## 方法2：

p<=A.tu

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | 3 | 4 | 4 | 6 |

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p=3

B

| | *i* | *j* | *e* |
|---|---|---|---|
| 0 | | | |
| 1 | | | |
| 2 | 2 | 1 | 14 |
| 3 | | | |
| 4 | | | |
| 5 | 5 | 1 | -5 |

## 方法2：

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | **3** | 4 | 4 | 6 |

**p<=A.tu**

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=3

B

| *i* | *j* | *e* | |
|---|---|---|---|
| | | | 0 |
| | | | 1 |
| 2 | 1 | 14 | 2 |
| | | | 3 |
| | | | 4 |
| 5 | 1 | -5 | 5 |

## 方法2：

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | 3 | 4 | 4 | 6 |

**p<=A.tu**

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=3

B

| | *i* | *j* | *e* |
|---|---|---|---|
| 0 | | | |
| 1 | | | |
| 2 | 2 | 1 | 14 |
| 3 | 2 | 2 | -7 |
| 4 | | | |
| 5 | 5 | 1 | -5 |

## 方法2：

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | **4** | 4 | 4 | 6 |

p<=A.tu

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=4

B

| | *i* | *j* | *e* |
|---|---|---|---|
| 0 | | | |
| 1 | | | |
| 2 | 2 | 1 | 14 |
| 3 | 2 | 2 | -7 |
| 4 | | | |
| 5 | 5 | 1 | -5 |

方法2：

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 1 | 4 | 4 | 4 | 6 |

**p<=A.tu**

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=4

B

| *i* | *j* | *e* | |
|---|---|---|---|
| | | | 0 |
| 1 | 3 | 36 | 1 |
| 2 | 1 | 14 | 2 |
| 2 | 2 | -7 | 3 |
| | | | 4 |
| 5 | 1 | -5 | 5 |

方法2：

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 2 | 4 | 4 | 4 | 6 |

p<=A.tu

A

| *i* | *j* | *e* | |
|---|---|---|---|
| | | | |
| 1 | 2 | 14 | |
| 1 | 5 | -5 | |
| 2 | 2 | -7 | |
| 3 | 1 | 36 | ← p=4 |
| 3 | 4 | 28 | |

B

| *i* | *j* | *e* | |
|---|---|---|---|
| | | | 0 |
| 1 | 3 | 36 | 1 |
| 2 | 1 | 14 | 2 |
| 2 | 2 | -7 | 3 |
| | | | 4 |
| 5 | 1 | -5 | 5 |

# 方法2：

p<=A.tu

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 2 | 4 | 4 | 4 | 6 |

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p=5

B

| *i* | *j* | *e* | |
|---|---|---|---|
| | | | 0 |
| 1 | 3 | 36 | 1 |
| 2 | 1 | 14 | 2 |
| 2 | 2 | -7 | 3 |
| | | | 4 |
| 5 | 1 | -5 | 5 |

# 方法2：

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 2 | 4 | 4 | 4 | 6 |

**p<=A.tu**

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

p=5

B

| | *i* | *j* | *e* |
|---|---|---|---|
| 0 | | | |
| 1 | 1 | 3 | 36 |
| 2 | 2 | 1 | 14 |
| 3 | 2 | 2 | -7 |
| 4 | 4 | 3 | 28 |
| 5 | 5 | 1 | -5 |

方法2：

| col | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Num[col] | 1 | 2 | 0 | 1 | 1 |
| Cpot[col] | 2 | 4 | 4 | 5 | 6 |

p<=A.tu

A

| *i* | *j* | *e* |
|---|---|---|
| | | |
| 1 | 2 | 14 |
| 1 | 5 | -5 |
| 2 | 2 | -7 |
| 3 | 1 | 36 |
| 3 | 4 | 28 |

← p=6

B

| | *i* | *j* | *e* |
|---|---|---|---|
| 0 | | | |
| 1 | 1 | 3 | 36 |
| 2 | 2 | 1 | 14 |
| 3 | 2 | 2 | -7 |
| 4 | 4 | 3 | 28 |
| 5 | 5 | 1 | -5 |

```
Status FastTransposeSMatrix(TSMatrix A, TSMatrix &B){
  B.mu = A.nu;  B.nu = A.mu;  B.tu = A.tu;
  if (B.tu) {
    for (col=1; col<=A.nu; ++col)  num[col] = 0;
    for (t=1; t<=A.tu; ++t)  ++num[A.data[t].j];
    cpot[1] = 1;
    for (col=2; col<=A.nu; ++col)
      cpot[col] = cpot[col-1] + num[col-1];
    for (p=1; p<=A.tu; ++p) {        ……        }
  } // if
  return OK;
} // FastTransposeSMatrix
```

```
Col = A.data[p].j;
q = cpot[col];
B.data[q].i = A.data[p].j;
B.data[q].j = A.data[p].i;
B.data[q].e = A.data[p].e;
++cpot[col]
```

时间复杂度为：O(A.nu+A.tu)

# 二、行逻辑联接的顺序表

三元组顺序表又称有序的双下标法，它的特点是，非零元在表中按行序有序存储，因此便于进行依行顺序处理的矩阵运算。然而，若需随机存取某一行中的非零元，则需从头开始进行查找。

**行逻辑联接的顺序表**:随机存取某一行中的非零元

```
#define  MAXRC  500
   typedef struct {
      Triple  data[MAXSIZE + 1];
       int     rpos[MAXRC + 1]; // 每一行非0元存放的起始位置

       int     mu, nu, tu;
   } RLSMatrix;   // 行逻辑链接顺序表类型
```

$$\begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix}$$

| $i$ | $j$ | $e$ | |
|---|---|---|---|
| | | | T.data[0] |
| **1** | **2** | **14** | **T.data[1]** |
| **1** | **5** | **-5** | **T.data[2]** |
| **2** | **2** | **-7** | **T.data[3]** |
| **3** | **1** | **36** | **T.data[4]** |
| **3** | **4** | **28** | **T.data[5]** |

rpos[];

| |
|---|
| |
| **1** |
| **3** |
| **4** |

## 例如：给定一组下标，求矩阵的元素值

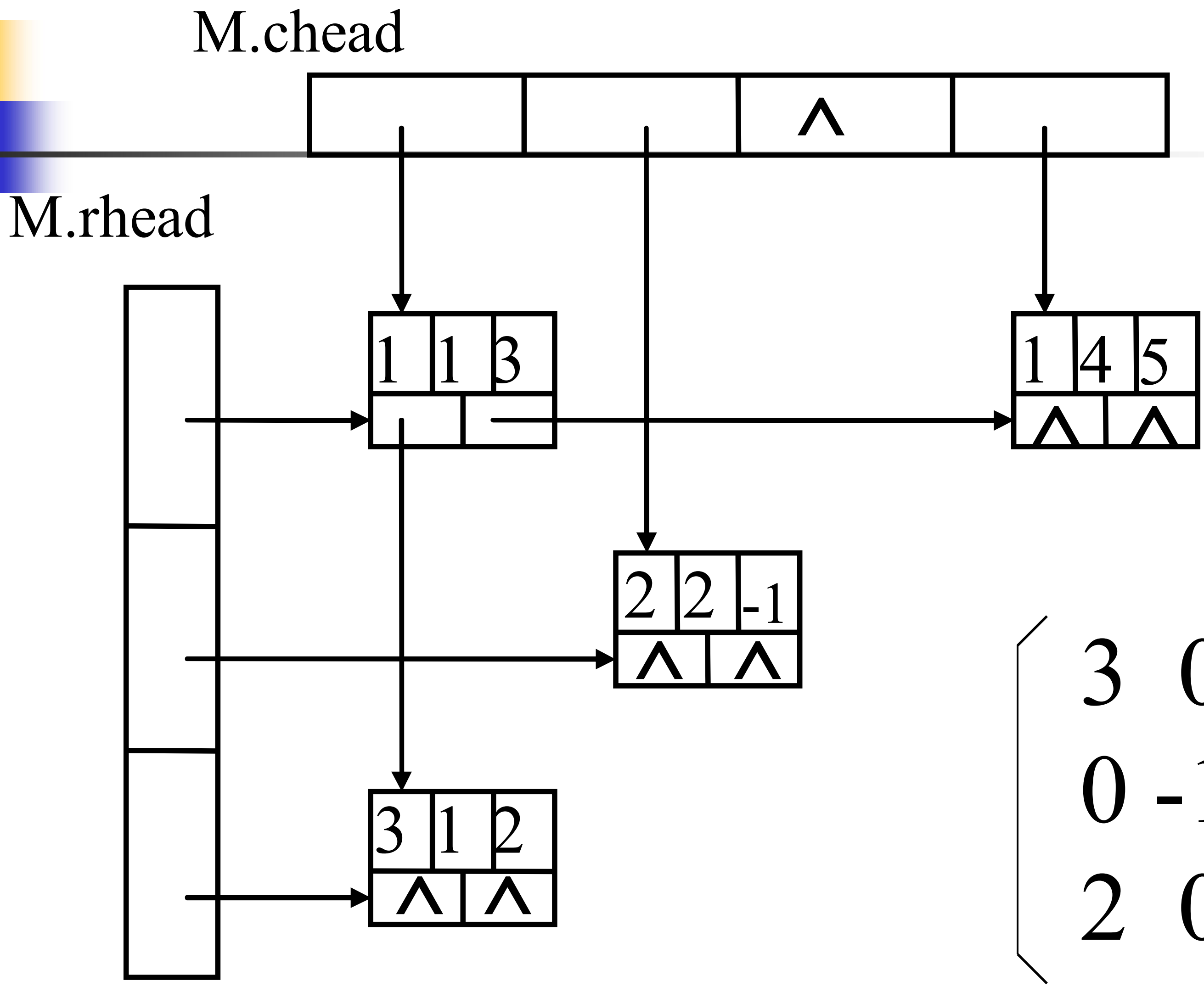
```
ElemType value(RLSMatrix M, int r, int c)
 {
   p = M.rpos[r];
   while (M.data[p].i==r &&M.data[p].j < c)
      p++;
   if (M.data[p].i==r && M.data[p].j==c)
     return M.data[p].e;
   else return 0;
} // value
```

# 三、 十字链表

- 采用链表存放稀疏矩阵的非0元
  - 将稀疏矩阵每行的非0元按照列升序的顺序放在一个单链表中
  - 将稀疏矩阵每列的非0元按照行升序的顺序放在一个单链表中
- 稀疏矩阵的每个非0元即位于一个行单链表，也同时位于一个列单链表
- 用一维数组保存每行非0元的单链表的头指针
- 用一维数组保存每列非0元的单链表的头指针
- 每个结点非0元的结点结构：
  - row，col,val分别代表非0元的行号，列号和值
  - down为指针，指向该非0元同一列的下一个非0元
  - right为指针，指向该非0元同一行的下一个非0元

<table>
<tr><td>row</td><td colspan="2">col</td><td>val</td></tr>
<tr><td colspan="2">down</td><td colspan="2">right</td></tr>
</table>

# 三、 十字链表




$$\begin{pmatrix} 3 & 0 & 0 & 5 \\ 0 & -1 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$$