第五章 数组与广义表

机电工程与自动化学院 L栋301

任卫红 助理教授

renweihong@hit.edu.cn

http://faculty.hitsz.edu.cn/renweihong

数组和广义表

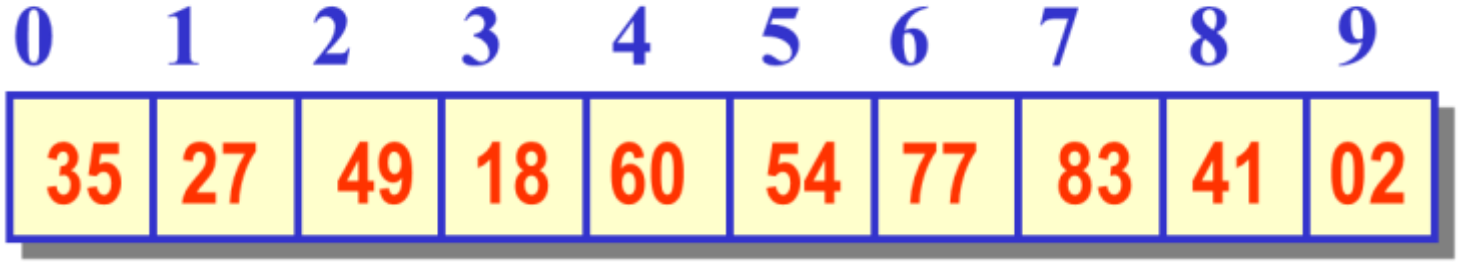
- 一维数组
- 多维数组
- 特殊矩阵的压缩存储
- 稀疏矩阵
- 广义表

5.1数组

第一节　数组的定义

一、数组

- 数组是相同类型的数据元素的集合
- 数组是一种定长的线性表
- 数组一般不作插入和删除操作
- 一旦建立了数组，则结构中的数据元素个数和数据元素之间的关系就不再发生变动



第一节　数组的定义

一、数组

- 数组中的每个数据元素都对应于一组下标
  $(j_1, j_2, \ldots, j_n)$
- 其中：$0 \leqslant j_i \leqslant b_i - 1$

  b_i称为第i维的长度$(i=1, 2, \ldots, n)$

判断：“数组的处理比其它复杂的结构要简单”，对吗？

答：对的。因为——

① 数组中各元素具有统一的类型；

② 数组元素的下标一般具有固定的上界和下界，即数组一旦被定义，它的维数和维界就不再改变。

③数组的基本操作比较简单，除了结构的初始化和销毁之外，只有存取元素和修改元素值的操作。

第一节　数组的定义

二、一维数组

- 一维数组是一种简单的定长线性表
- 一维数组中的每个数据元素是一个(数)值(原子)
- 如：int A[8]={8,7,5,4,6,1,3,2}

  b=8，有8个数据元素，每个元素都是一个数值

0	1	2	3	4	5	6	7
8	7	5	4	6	1	3	2

第一节　数组的定义

三、二维数组

- 二维数组是这样一个定长线性表，其每个数据元素也是一个定长线性表(一维数组)

$$A_{mxn} = \begin{pmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0,n-1} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m-1,0} & a_{m-1,1} & a_{m1,2} & \cdots & a_{m-1,n-1} \end{pmatrix}$$

- Amxn= ((a00 a01…a0,n-1), (a10 a11…a1,n-1),…,(am-1,0 …am-1,n-1))

第一节　数组的定义

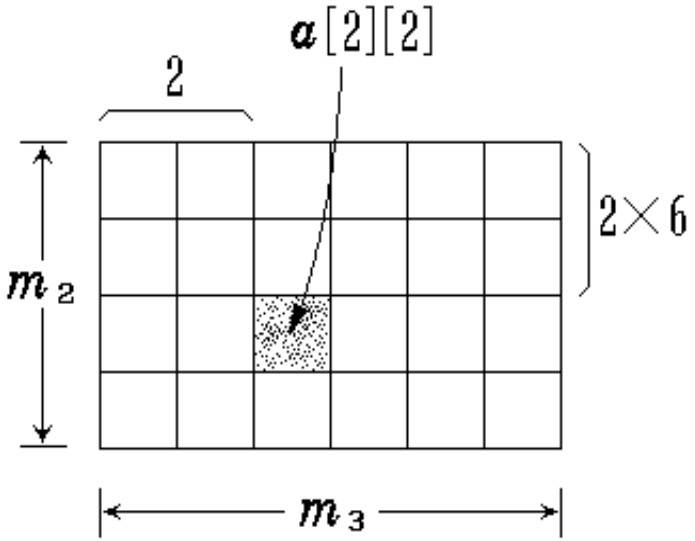
三、多维数组

- 多维数组是这样一个定长线性表，其每个数据元素也是一个定长线性表(降一维)
- 如果其数据元素不是一维数组，则其数据元素的每个数据元素也是一个定长线性表
- 一直到最后一个定长线性表是一维数组，其每个数据元素为一个(数)值

第一节　数组的定义

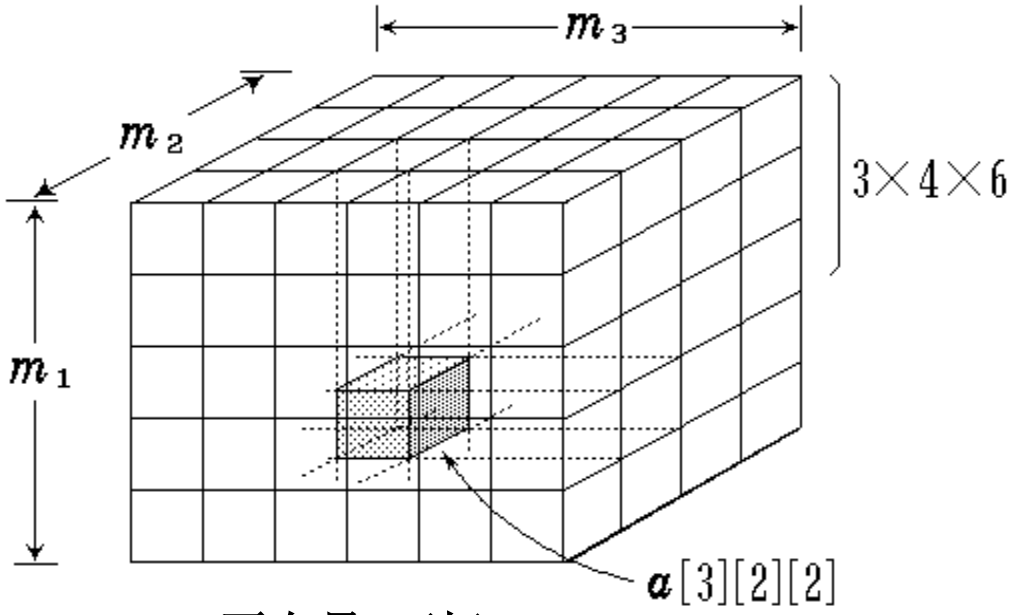
二维数组

$m_1=5$　$m_2=4$　$m_3=6$



行向量　下标 i
列向量　下标 j

三维数组



页向量　下标 i
行向量　下标 j
列向量　下标 k

5.2 数组的表示

第二节　数组的表示

一、数组的顺序表示

- **顺序存储**：数组由相同类型的数据组成，且一般不作插入和删除操作，一般采用顺序存储结构表示数组
- 次序约定：**计算机中，存储单元是一维结构**，而数组为多维结构，则用一组连续的存储单元存放数组的数据元素时，有一个**次序约定**问题

第二节　数组的表示

二、二维数组的顺序表示

- A_{mxn} =

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0,n-1} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m-1,0} & a_{m-1,1} & a_{m1,2} & \cdots & a_{m-1,n-1} \end{pmatrix}$$

行序

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0,n-1} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m-1,0} & a_{m-1,1} & a_{m1,2} & \cdots & a_{m-1,n-1} \end{pmatrix}$$

列序

- A_{mxn}= $((a_{00}\ a_{01}\ldots a_{0,n-1}), (a_{10}\ a_{11}\ldots a_{1,n-1}),\ldots,(a_{m-1,0}\ \ldots a_{m-1,n-1}))$
- A_{mxn}= $((a_{00}\ a_{10}\ldots a_{m-1,0}), (a_{01}\ a_{11}\ldots a_{m-1,1}),\ldots,(a_{0,n-1}\ \ldots a_{m-1,n-1}))$

第二节　数组的表示

二、二维数组的顺序表示

- A_{mxn}以行序为主序存储

$$A_{mxn}=\begin{pmatrix} a_{00} & a_{01} & a_{02} & \cdots & a_{0,n-1} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m-1,0} & a_{m-1,1} & a_{m1,2} & \cdots & a_{m-1,n-1} \end{pmatrix}$$

a_{00}	a_{01}	…	$a_{0,\,n-1}$	a_{10}	a_{11}	…	$a_{1,\,n-1}$	…	$a_{m-1,\,0}$	…	$a_{m-1,\,n-1}$

- $LOC(a_{ij}) = LOC(a_{00}) + (i \times n + j) \times L$
- $LOC(a_{00})$是二维数组的起始存储地址
- L为每个数据元素占用存储单元的长度(数目)

第二节　数组的表示

二、二维数组的顺序表示

- A_{mxn}以列序为主序存储

$$A_{mxn}=\left(\begin{pmatrix}a_{00}\\a_{10}\\\vdots\\a_{m-1,0}\end{pmatrix}\begin{pmatrix}a_{01}\\a_{11}\\\vdots\\a_{m-1,1}\end{pmatrix}\begin{pmatrix}a_{02}\\a_{12}\\\vdots\\a_{m1,2}\end{pmatrix}\cdots\begin{pmatrix}a_{0,n-1}\\a_{1,n-1}\\\vdots\\a_{m-1,n-1}\end{pmatrix}\right)$$

a_{00}	a_{10}	…	$a_{m-1,0}$	a_{01}	a_{11}	…	$a_{m-1,1}$	…	$a_{0,n-1}$	…	$a_{m-1,n-1}$

- $LOC(a_{ij}) = LOC(a_{00}) + (i + j \times m) \times L$
- $LOC(a_{00})$是二维数组的起始存储地址
- L为每个数据元素占用存储单元的长度(数目)

例1：一个二维数组A，行下标的范围是1到6，列下标的范围是0到7，每个数组元素用相邻的6个字节存储，存储器按字节编址。那么，这个数组的体积是_____288_____个字节。

答：Volume=m*n*L=(6-1+1)*(7- 0 +1)*6=48*6=288

例2：已知二维数组$A_{m,m}$按行存储的元素地址公式是：$Loc(a_{ij})= Loc(a_{11})+[(i-1)*m+(j-1)]*K$ ， 请问按列存储的公式相同吗？

答：尽管是方阵，但公式仍不同。应为：

$Loc(a_{ij})=Loc(a_{11})+[(j-1)*m+(i-1)]*K$

例3 ：〖考研题〗 ：设数组a[1…60，1…70]的基地址为2048，每个元素占2个存储单元，若以列序为主序顺序存储，则元素a[32,58]的存储地址为_____8950_____。

答：请注意审题！

根据列优先公式 $Loc(a_{ij})=Loc(a_{11})+[(j-1)*m+(i-1)]*K$

得：$LOC(a_{32,58})=2048+[(58-1)*60+(32-1)]*2=8950$

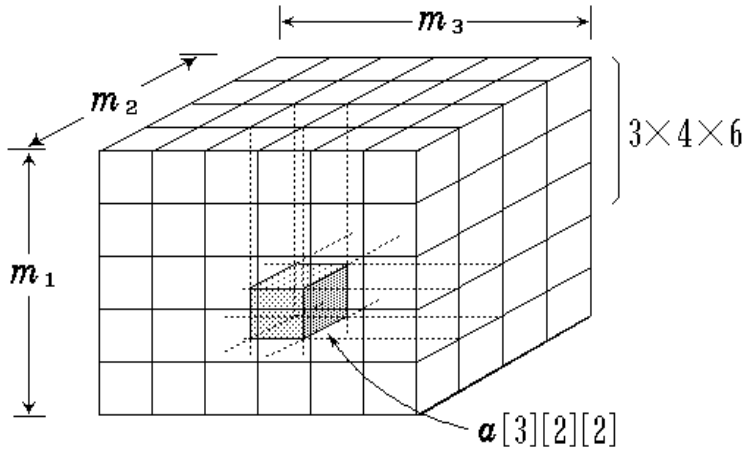
想一想：若数组是a[0…59，0…69]，结果是否仍为8950？

维界虽未变，但此时的a[32,58]不再是原来的a[32,58]

第二节　数组的表示

三、三维数组的顺序表示（按行）

- 各维元素个数为 m_1, m_2, m_3
- 下标为 i_1, i_2, i_3的数组元素的



(按页 / 行 / 列存放)

$$LOC(i_1, i_2, i_3) = a + (i_1 * m_2 * m_3 + i_2 * m_3 + i_3) * l$$

$i_1 * m_2 * m_3$：前i_1页总元素个数

$i_2 * m_3$：第i_1页前i_2行总元素个数

i_3：第 i_2 行前 i_3 列元素个数

第二节　数组的表示

三、多维数组的顺序表示

- 以行序为主序存储，多(K)维数组元素存储位置
- $LOC(a_{j1,j2,\ldots,jk}) = LOC(a_{00\ 0}) + ((b_2 \times b_3 \times \ldots \times b_k \times j_1) + (b_3 \times \ldots \times b_k \times j_2) + \ldots + j_k) \times L$

$a_{0\ 00}$	$a_{0\ 01}$	$\cdots$	$a_{0\ 0bk-1}$	$a_{0\ 10}$	$a_{0\ 11}$	$\cdots$	$a_{0,1bk-1}$	$\cdots$	$a_{b1-1,\ ,0}$	$\cdots$	$a_{b1-1,,bk-1}$

第二节　数组的表示

三、多维数组的顺序表示

- 以列序为主序存储，多(K)维数组元素存储位置
- $LOC(a_{j1,j2,\ldots,jk}) = LOC(a_{00\ 0}) + ((b_1 \times b_2 \times \ldots \times b_{k-1} \times j_k) + (b_1 \times \ldots \times b_{k-2} \times j_{k-1}) + \ldots + j_1) \times Lp$

$a_{00\ 0}$	$a_{10\ 0}$	$\cdots$	$a_{b1-1,0\ 0}$	$a_{01\ 0}$	$a_{11\ 0}$	$\cdots$	$a_{b1-1,1,0}$	$\cdots$	$a_{0\ b2-1,b1-1}$	$\cdots$	$a_{b1-1,,bk-1}$

5.3 矩阵的压缩存储

第三节　矩阵的压缩存储

一、矩阵的压缩存储

- 如果矩阵中有许多值相同的元素或者零元素(特殊矩阵、稀疏矩阵)，为了节省存储空间，可以对这类矩阵进行压缩存储
- 压缩存储：为多个值相同的元素只分配一个存储空间；对零元素不分配空间

第三节　矩阵的压缩存储

二、特殊矩阵

- 特殊矩阵：矩阵中，值相同的元素或者零元素的分布有一定规律
- 对称矩阵：矩阵中，对角线两边对应位置上元素的值相同（$a_{ij}=a_{ji}$）

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

第三节　矩阵的压缩存储

二、特殊矩阵

- 三角矩阵：矩阵中，对角线上（下）边元素值为常数（或者0），称下（上）三角矩阵

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

下三角矩阵

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

上三角矩阵

第三节　矩阵的压缩存储

二、特殊矩阵(对称矩阵的压缩存储)

- 设有一个 n×n 的对称矩阵 A

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

在矩阵中，$a_{ij} = a_{ji}$

第三节　矩阵的压缩存储

- 为节约存储，只存对角线及对角线以上的元素，或者只存对角线或对角线以下的元素。前者称为上三角矩阵，后者称为下三角矩阵。

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

下三角矩阵

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

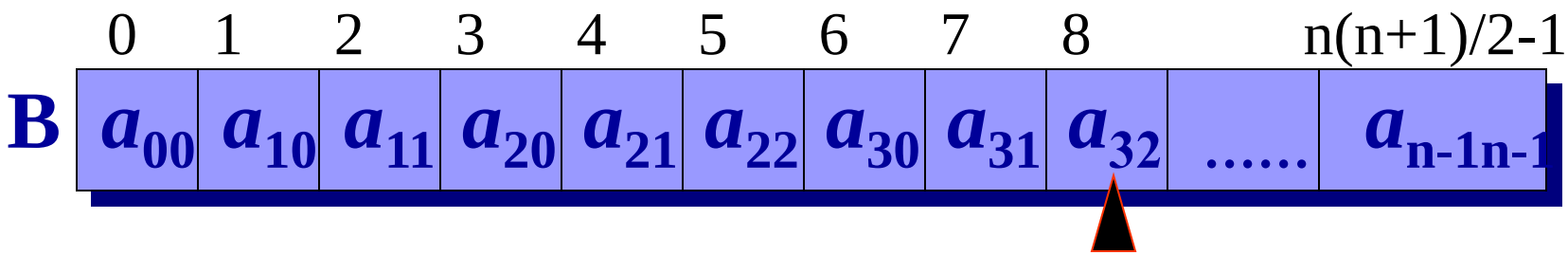
上三角矩阵

- 把它们按行存放于一个一维数组 B 中，称之为对称矩阵 A 的压缩存储方式。
- 数组 B 共有 n + (n - 1) + … + 1 =

  n*(n+1)/2 个元素。

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

下三角矩阵



若 $i \geq j$, 数组元素A[i][j]在数组B中的存放位置为 1 + 2 + … + i + j = (i + 1)* i / 2 + j

前i行元素总数　第i行第j个元素前元素个数

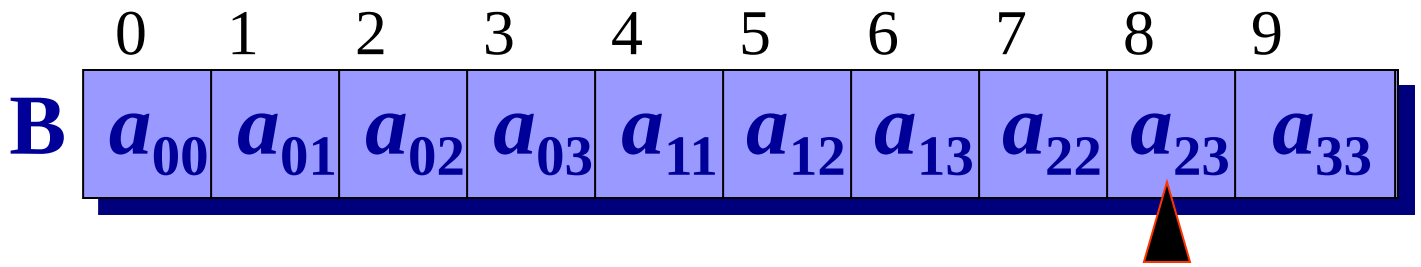
$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & \Lambda & a_{0n-1} \\ a_{10} & a_{11} & a_{12} & \Lambda & a_{1n-1} \\ a_{20} & a_{21} & a_{22} & \Lambda & a_{2n-1} \\ \Lambda & \Lambda & \Lambda & \Lambda & \Lambda \\ a_{n-10} & a_{n-11} & a_{n-12} & \Lambda & a_{n-1n-1} \end{bmatrix}$$

下三角矩阵

若 $i < j$，数组元素 A[*i*][*j*] 在矩阵的上三角部分，在数组 B 中没有存放，可以找它的对称元素A[*j*][*i*]：= *j* *(*j* +1) / 2 + *i*

上三角矩阵

$$n = 4 \quad \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$



若$i \le j$，数组元素A[i][j]在数组B中的存放位置为

$$n + (n-1) + (n-2) + \cdots + (n-i+1) + j-i$$

前i行元素总数　第i行第j个元素前元素个数

若 $i \le j$，数组元素A[i][j]在数组B中的存放位置为

$$n + (n-1) + (n-2) + \cdots + (n-i+1) + j-i =$$
$$= (2*n-i+1) * i / 2 + j-i =$$
$$= (2*n-i-1) * i / 2 + j$$

若$i > j$，数组元素A[i][j]在矩阵的下三角部分，在数组 B 中没有存放。因此，找它的对称元素A[j][i]。

A[j][i]在数组 B 的第 $(2*n-j-1) * j / 2 + i$ 的位置中找到。

第三节　矩阵的压缩存储

三、稀疏矩阵

- 稀疏矩阵：矩阵中有许多值相同的元素或者零元素，而且分布没有任何规律
- 假设在m×n的矩阵中，有t个非零元素，令：

$$\delta = t /(m \times n)$$

- 如果稀疏因子 $\delta \leqslant 0.05$，则称该矩阵为稀疏矩阵

第三节　矩阵的压缩存储

三、稀疏矩阵

- 稀疏矩阵：矩阵中有许多值相同的元素或者零元素，而且分布没有任何规律

$$A_{6\times7}=\begin{pmatrix}0&0&0&22&0&0&15\\0&11&0&0&0&17&0\\0&0&0&-6&0&0&0\\0&0&0&0&0&39&0\\91&0&0&0&0&0&0\\0&0&28&0&0&0&0\end{pmatrix}$$

第三节　矩阵的压缩存储

三、稀疏矩阵（三元组）

- 用三元组存储稀疏矩阵中的**非零元素**
- 三元组(i, j, a_{ij})表示矩阵中i行、j列位置的值为a_{ij}

$$\begin{pmatrix} 0 & 0 & 0 & 22 & 0 & 0 & 15 \\ 0 & 11 & 0 & 0 & 0 & 17 & 0 \\ 0 & 0 & 0 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 39 & 0 \\ 91 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 0 & 0 & 0 \end{pmatrix}$$

	行 (row)	列 (col)	值 (value)
[0]	0	3	22
[1]	0	6	15
[2]	1	1	11
[3]	1	5	17
[4]	2	3	−6
[5]	3	5	39
[6]	4	0	91
[7]	5	2	28

第三节　矩阵的压缩存储

三、稀疏矩阵(转置)

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 91 & 0 \\ 0 & 11 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 28 \\ 22 & 0 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 17 & 0 & 39 & 0 & 0 \\ 15 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

	行 (row)	列 (col)	值 (value)
[0]	0	4	91
[1]	1	1	11
[2]	2	5	28
[3]	3	0	22
[4]	3	2	−6
[5]	5	1	17
[6]	5	3	39
[7]	6	0	16

第三节　矩阵的压缩存储

三、稀疏矩阵(转置)

原矩阵三元组表

	行 (row)	列 (col)	值 (value)
[0]	0	3	22
[1]	0	6	15
[2]	1	1	11
[3]	1	5	17
[4]	2	3	−6
[5]	3	5	39
[6]	4	0	91
[7]	5	2	28

转置矩阵三元组表

	行 (row)	列 (col)	值 (value)
[0]	0	4	91
[1]	1	1	11
[2]	2	5	28
[3]	3	0	22
[4]	3	2	−6
[5]	5	1	17
[6]	5	3	39
[7]	6	0	16

第三节　矩阵的压缩存储

三、稀疏矩阵(转置)

	行 (row)	列 (col)	值 (value)
[0]	0	3	22
[1]	0	6	15
[2]	1	1	11
[3]	1	5	17
[4]	2	3	−6
[5]	3	5	39
[6]	4	0	91
[7]	5	2	28

	行 (row)	列 (col)	值 (value)
[0]	0	4	91
[1]	1	1	11
[2]	2	5	28
[3]	3	0	22
[4]	3	2	−6
[5]	5	1	17
[6]	5	3	39
[7]	6	0	16

- 设矩阵列数为 Cols，对矩阵三元组表扫描Cols 次。
- 第 k 次扫描找寻所有列号为 k 的项，将其行号变列号、列号变行号，顺次存于转置矩阵三元组表。

第四节　广义表的定义

一、广义表的定义

- 广义表：由n（≥0）个表元素组成的有限序列：

$$LS = (a_0, a_1, a_2, \ldots, a_{n-1})$$

- LS是广义表的名称
- a_i是广义表的元素，既可以是表（称为子表），也可以是数据元素（称为原子）
- n为广义表的长度（n=0的广义表为空表）

第四节　广义表的定义

二、广义表举例

- A=( );　　//表A是一个空表
- B=(e);　　//表B有一个原子
- C=(a, (b, c, d));　　//两个元素，分别为原子a和子表(b, c, d)
- D=(A, B, C);　　//有三个元素均为列表
- E=(a, E);　　//递归的列表

其中，“表”以及“列表”，均指广义表

第四节　广义表的定义

三、广义表的表头

- 表头(head)：广义表的第一个元素
- 表头既可以是原子，也可以是列表(广义表)
- GetHead(B) = e
- GetHead(D) = A
- GetHead((B,C)) = B

A=( );
B=(e);
C=(a, (b, c, d));

D=(A, B, C);
E=(a, E);

第四节　广义表的定义

四、广义表的表尾

- 表尾(tail)：广义表中，除表头外的部分
- 表尾一定是列表
- GetTail(B) = ()
- GetTail(D) = (B,C)
- GetTail((B,C)) = (C)

A=( );
B=(e);
C=(a, (b, c, d));

D=(A, B, C);
E=(a, E);

例：求下列广义表操作的结果

1. GetTail [(b, k, p, h)]＝ (k, p, h) ;

2. GetHead [((a,b), (c,d))]＝ (a,b) ;

3. GetTail [((a,b), (c,d))]＝ ((c,d)) ;

4. GetTail [GetHead[((a,b),(c,d))]]＝（b） ;

5. GetTail [（e）]＝ ( ) ;

6. GetHead [(( ))]＝ ( ) .

7. GetTail [(( ))]＝ ( ) .

(a,b)

第五节　广义表的存储

一、广义表的存储结构

- 广义表一般采用链式存储结构
- 表结点

Tag=1	hp	tp

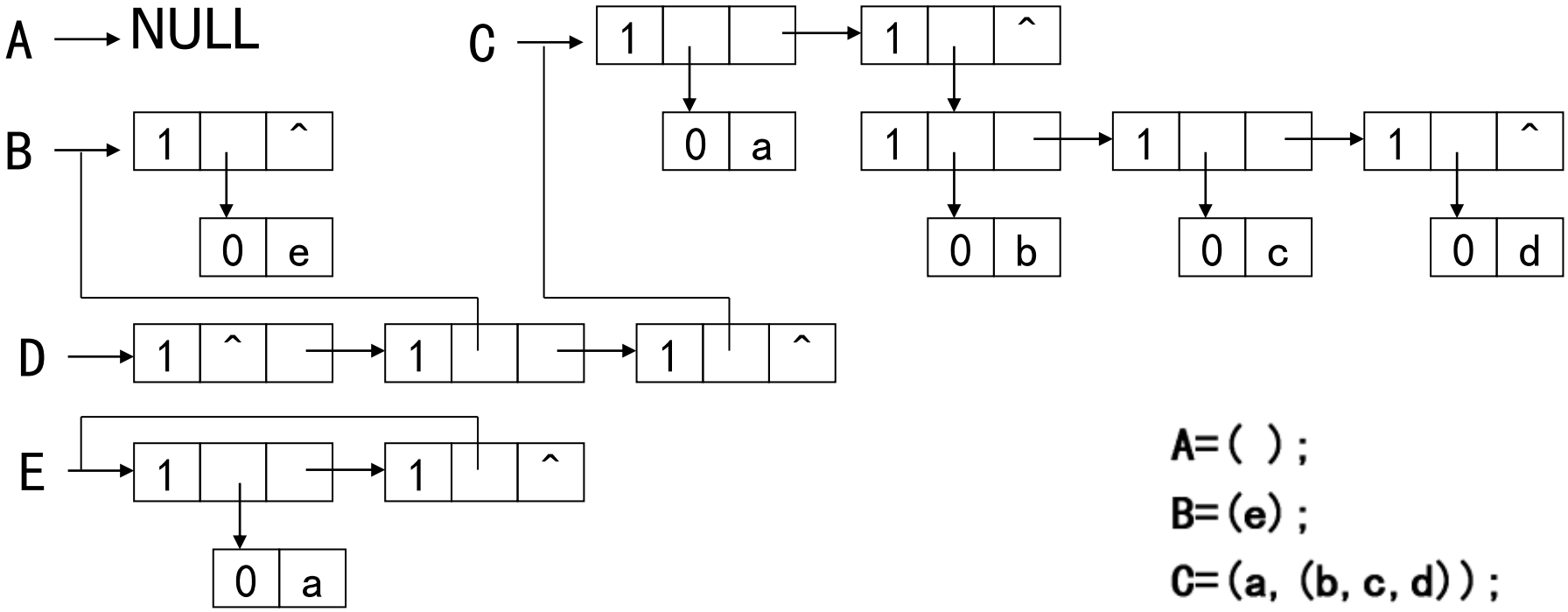
- 原子结点

Tag=0	atom

- hp表示表头，tp表示表尾

第五节　广义表的存储

一、广义表的存储结构



A=( );

B=(e);

C=(a, (b, c, d));

D=(A, B, C);

E=(a, E);

作业

1. 假设一个10×10的上三角矩阵A按照列优先顺序压缩存储在一维数组 B中，则B数组的大小应为

- A. 50　　B. 55　C. 100　D. 101

作业

2. A是7×4的二维数组，按行优先方式顺序存储，元素A[0][0]的存储地址为1000，若每个元素占2个字节，则A[3][3]的存储地址为

A. 1026　　B. 1028　　C. 1030　　D. 1032

作业

3. 对稀疏矩阵采用三元组表示法的目的是

- A. 便于输入和输出
- B. 便于进行矩阵运算
- C. 降低时间复杂度
- D. 节省存储空间

作业

4. 已知一个5×5的稀疏矩阵所示，试写出该矩阵的三元组表示。

$$\begin{pmatrix} 5 & 19 & 0 & 0 & 0 \\ 18 & 70 & 0 & 0 & 0 \\ 3 & 3 & 4 & 1 & 2 \\ 6 & 9 & 14 & 11 & 10 \\ 5 & 4 & 8 & 4 & 6 \end{pmatrix}$$

作业

5. 一个数组的第一个元素的存储地址是100，每个元素占2存储单元，则第5个元素的存储地址是_____m

A. 105　B.108　C. 115　D. 118

作业

6. 假设一个8阶的上三角矩阵A按照列优先顺序压缩存储在一维数组8中，则B数组的大小应为_____。

作业

7. 二维数组A[8][9]按行优先顺序存储，若数组元素A[2][3]的存储地址为l087，A[4][7]的存储地址为ll53，则每个数组元素占用的存储单元的个数是_____

作业

8. 二维数组A按行序优先顺序存储，每个数据元素占1个存储单元。若数据元素 A[1][1]的存储地址是420，A[3][3]的存储地址是446，则A[5][5]的存储地址是

A．470　　B. 471　　C．472　　D．473

作业

9. 稀疏矩阵可以采用_____方法进行压缩存储。

作业

10. 把特殊矩阵A[10][10]的下三角矩阵压缩存储到一个一维数组M中，刚A中元素a[4][3]在M中所对应的下标位置是

A. 8　　B. 12　　C. 13　　D. 55

作业

11. 设有二维数组A[8][10]，按行序优先存储，且每个元素占用2个存储单元，若第一个元素的存储起始位置为b，则存储位置为b+20处的元素为_____。