数据结构

（C语言版）（第2版）

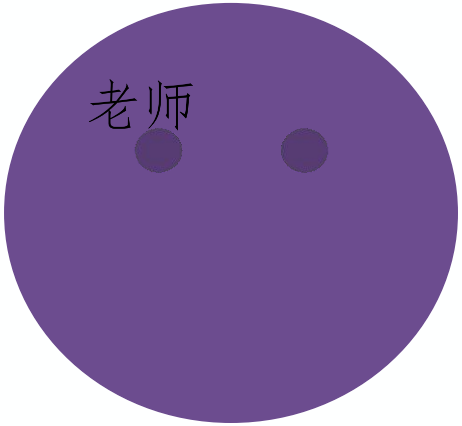
串、数组和广义表

数组与广义表

主讲教师：汪红松

教 学 内 容 Contents

一、数组的抽象数据类型
二、数组
三、特殊矩阵的压缩存储
四、广义表



一、数组的抽象数据类型

ADT Array {

数据对象:

$$j_i = 0, \cdots b_i - 1,\ \mathrm{i} = 1,2, \ \cdots \ , \mathrm{n}$$

$$D = \{\mathrm{a}_{\mathrm{j}\ \mathrm{j}_2 \cdots \mathrm{j}_\mathrm{n}} \mid \mathrm{a}_{\mathrm{j}_1 \mathrm{j}_2 \cdots \mathrm{j}_\mathrm{n}} \in ElemSet$$

数据关系:

$$R_1 = \{< a_{j_1 \cdots j_i \cdots j_n}, a_{j_1 \cdots j_i + 1 \cdots j_n} > |$$

$$0 \le \mathrm{j_k} \le b_k - 1,\ \ 1 \le \mathrm{k} \le \mathrm{n},\ \ \text{且}\mathrm{k} \ne \mathrm{i},$$

$$0 \le \mathrm{j_i} \le b_k - 2,$$

$$a_{j_1 \cdots j_i \cdots j_n}, a_{j_1 \cdots j_i + 1 \cdots j_n} \in D, i = 2, \cdots, n\}$$

一、数组的抽象数据类型

基本操作:

(1) InitArray (&A,n,bound1, …, boundn)

//构造数组A

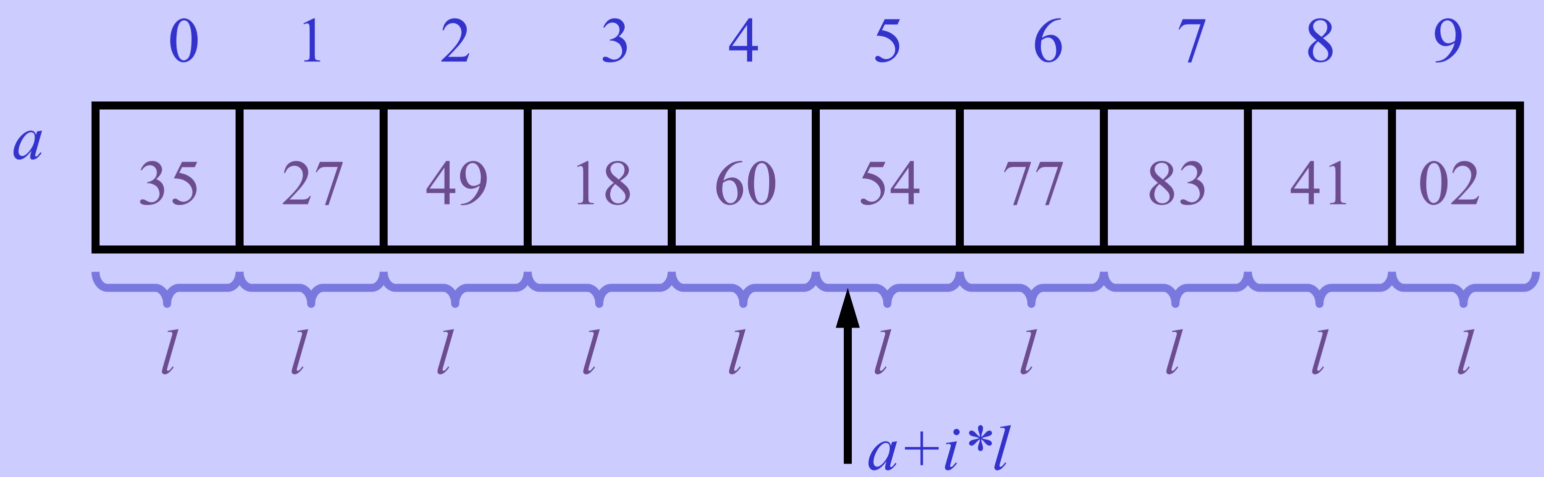
(2) DestroyArray (&A) // 销毁数组A

(3) Value(A,&e,index1,…,indexn) //取数组元素值

(4) Assign (A,&e,index1,…,indexn) //给数组元素赋值

}ADT Array

$$\mathrm{LOC}(i) = \begin{cases} a, & i = 0 \\ \mathrm{LOC}(i\text{-}1)+l = a+i*l, & i > 0 \end{cases}$$



$$\mathrm{LOC}(i) = \mathrm{LOC}(i\text{-}1)+l = a+i*l$$

$$A=(\alpha_1,\alpha_2,\cdots,\alpha_p)\quad (\mathrm{p}=\mathrm{m}或\mathrm{n})$$

$$\alpha_i=(a_{i1},a_{i2},\cdots,a_{in})\quad 1\le i\le m$$

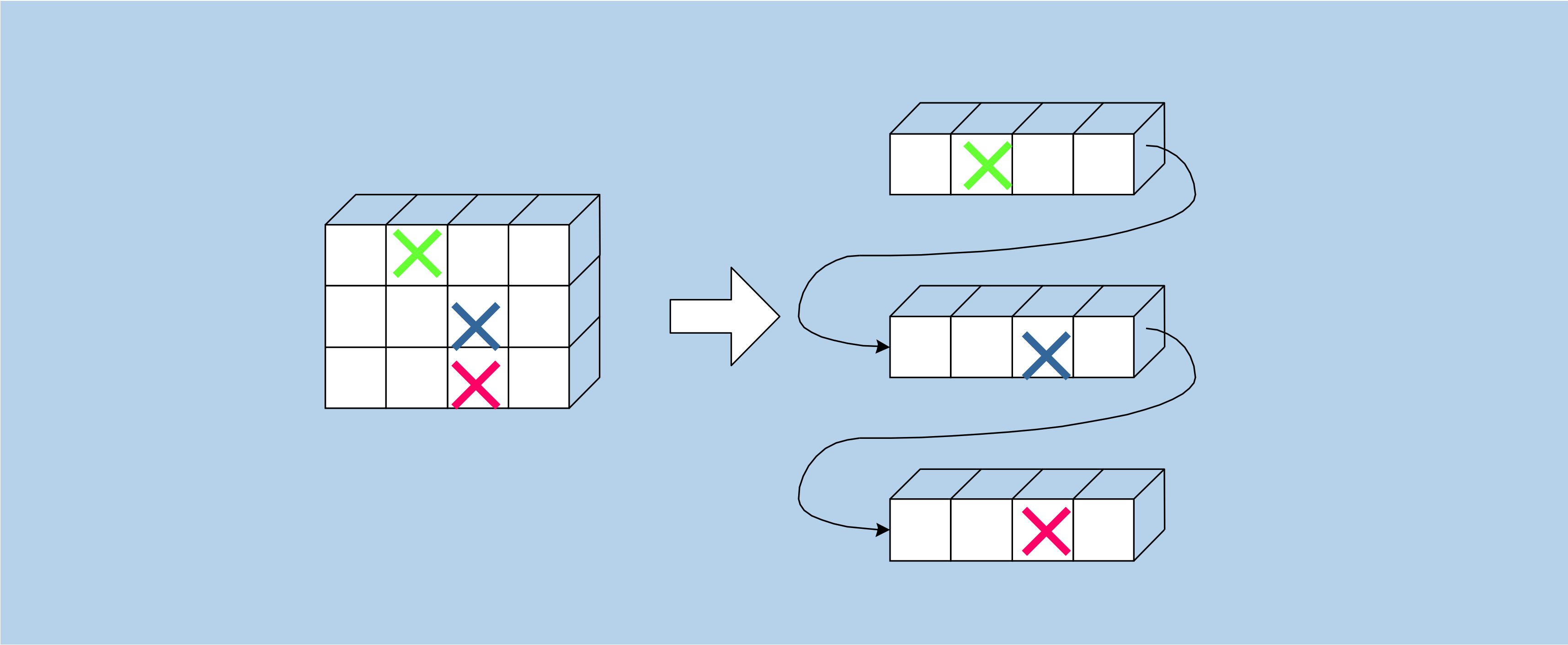
$$A_{m\times n}=\begin{bmatrix} [a_{11} & a_{12} & \cdots & a_{1n}] \\ [a_{21} & a_{22} & \cdots & a_{2n}] \\ \vdots & \vdots & & \vdots \\ [a_{m1} & a_{m2} & \cdots & a_{mn}] \end{bmatrix}$$

$$\alpha_j=(a_{1j},a_{2j},\cdots,a_{mj})\quad 1\le \mathrm{j}\le n$$

$$A_{m\times n}=\left[\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} \begin{matrix} \cdots \\ \cdots \\ \\ \cdots \end{matrix} \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}\right]$$

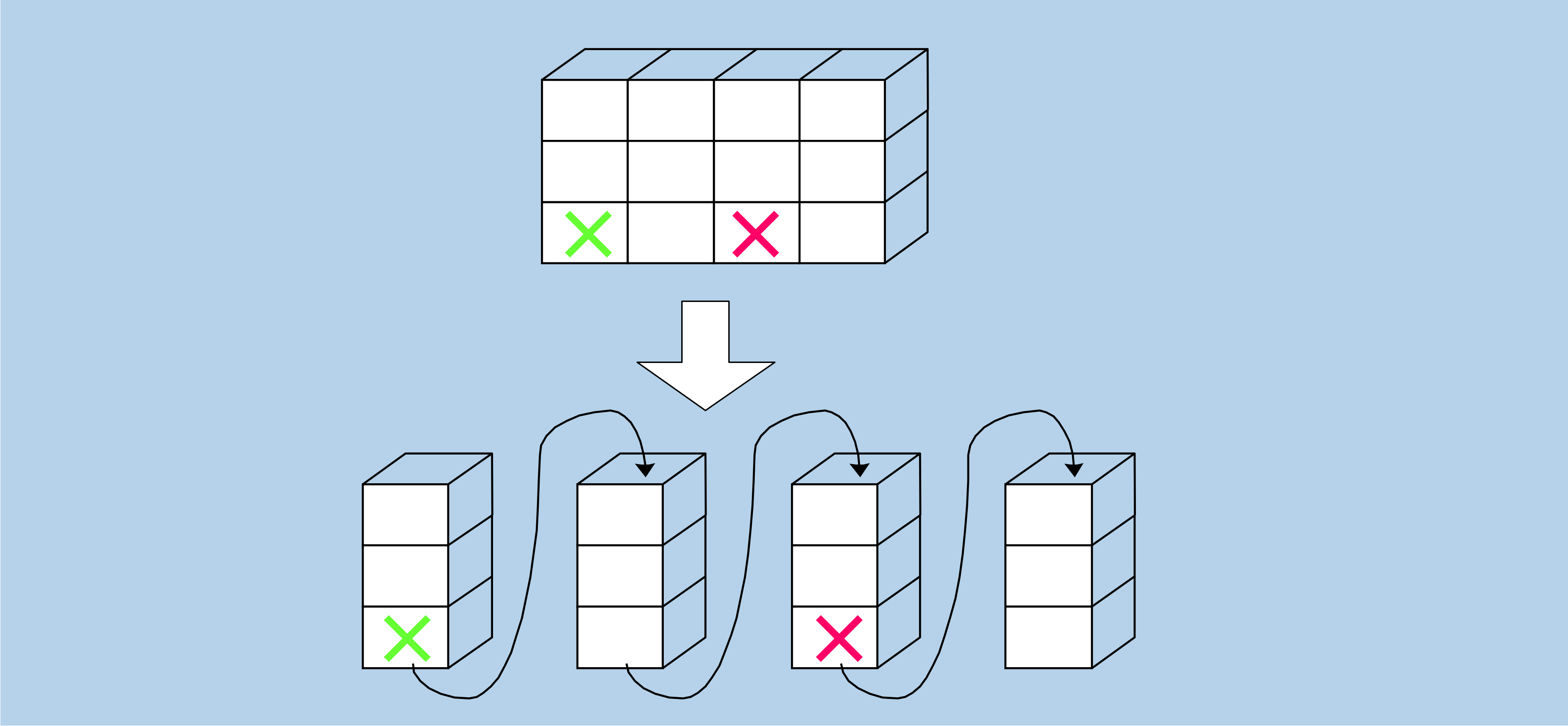
数组的顺序存储

- **以行序为主序** C, PASCAL



数组的顺序存储

- 以列序为主序 FORTRAN



（1）二维数组的行序优先表示

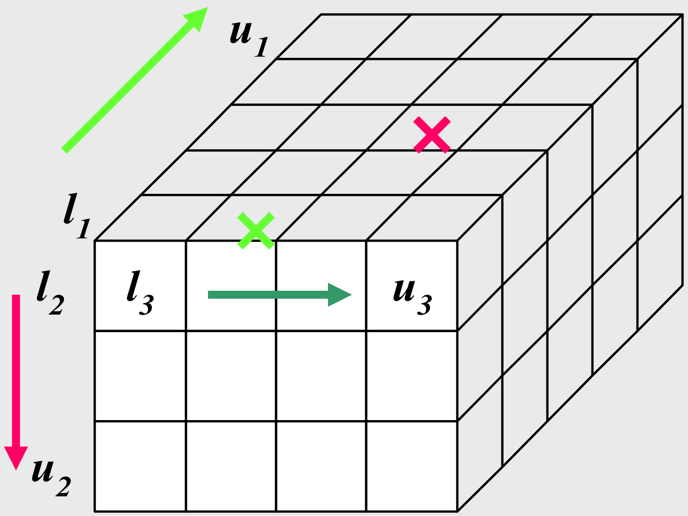
a[n][m]

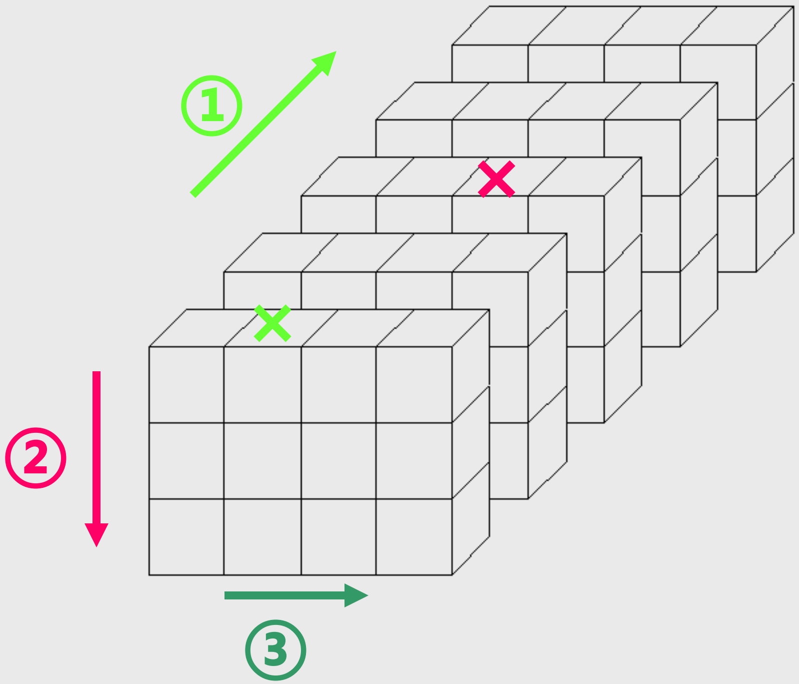
$$
a = \begin{pmatrix}
a[0][0] & a[0][1] & \cdots & a[0][m-1] \\
a[1][0] & a[1][1] & \cdots & a[1][m-1] \\
a[2][0] & a[2][1] & \cdots & a[2][m-1] \\
\vdots & \vdots & \ddots & \vdots \\
a[n-1][0] & a[n-1][1] & \cdots & a[n-1][m-1]
\end{pmatrix}
$$

设数组开始存放位置 $\mathrm{LOC}(0,0) = a$

$$\mathrm{LOC}(j,k) = a + (j * m + k) * l$$

按页/行/列存放，页优先的顺序存储





☞ a[m1][m2] [m3] 各维元素个数为 m_1, m_2, m_3

☞ 下标为 i_1, i_2, i_3的数组元素的存储位置：

$$\mathrm{LOC}(i_1, i_2, i_3) = a + (\underbrace{i_1 * m_2 * m_3}_{\text{前}i_1\text{页总元素个数}} + \underbrace{i_2 * m_3}_{\text{第}i_1\text{页的前}i_2\text{行总元素个数}} + i_3) *1$$

第 i_2 行前 i_3 列元素个数

- 各维元素个数为 $m_1, m_2, m_3, \ldots, m_n$
- 下标为 $i_1, i_2, i_3, \ldots, i_n$ 的数组元素的存储位置：

$$LOC(i_1, i_2, \cdots, i_n) = \mathrm{a} + i_1 * m_2 * m_3 * \cdots * m_n +$$

$$+ i_2 * m_3 * m_4 * \cdots * m_n + \cdots\cdots + i_{n-1} * m_n + i_n$$

$$= \mathrm{a} + \left(\left(\sum_{j=1}^{n-1} i_j * \prod_{k=j+1}^{n} m_k \right) + i_n \right) * L$$

4.n维数组

$$LOC[j_1, j_2, \cdots, j_n] = LOC[0,0,\cdots,0] + \left(\sum_{i=1}^{n} c_i j_i\right) L$$

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

三、特殊矩阵的压缩存储

1. **什么是压缩存储？**

   若多个数据元素的值都相同，则只分配一个元素值的存储空间，且零元素不占存储空间。

2. **什么样的矩阵能够压缩？**

   一些特殊矩阵，如：对称矩阵，对角矩阵，三角矩阵，稀疏矩阵等。

3. **什么叫稀疏矩阵？**

   矩阵中非零元素的个数较少（一般小于5%）

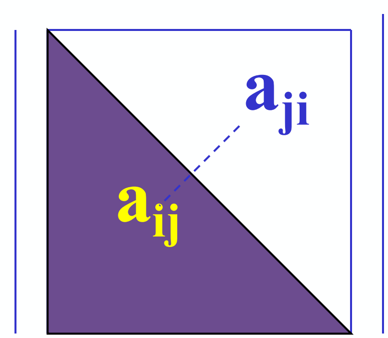
[特点] 在n*n的矩阵a中，满足如下性质：

$$a_{ij}=a_{ji} \quad (1 \le i, j \le n)$$

[存储方法] 只存储下(或者上)三角(包括主对角线)的数据元素。共占用n(n+1)/2个元素空间。

sa	a_{11}	a_{21}	a_{22}	a_{31}	…	$a_{ij}(a_{ji})$	…	a_{nn}
k	1	2	3	4				n(n+1)/2

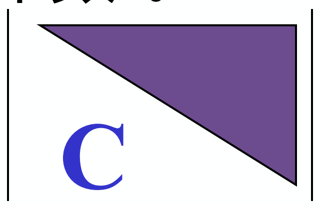
$$k=\begin{cases} i(i-1)/2+j & \text{当}i\ge j \\ j(j-1)/2+i & \text{当}i<j \end{cases}$$



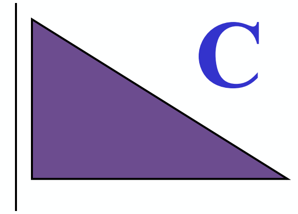
三、特殊矩阵的压缩存储

2. 三角矩阵

[特点] 对角线以下(或者以上)的数据元素(不包括对角线)全部为常数c。



上三角矩阵



下三角矩阵

[存储方法] 重复元素c共享一个元素存储空间，共占用 $n(n+1)/2+1$个元素空间

上三角矩阵

$$k=\begin{cases}(i-1)*(2n-i+2)/2+j-i+1 & i\le j\\ n(n+1)/2+1 & i>j\end{cases}$$

下三角矩阵

$$k=\begin{cases}i*(i-1)/2+j & i\ge j\\ n(n+1)/2+1 & i<j\end{cases}$$

三、特殊矩阵的压缩存储 3. 对角矩阵（带状矩阵）

[特点] 在n*n的方阵中，非零元素集中在主对角线及其两侧共L(奇数)条对角线的带状区域内 — L对角矩阵。

[存储方法]

以对角线的顺序存储

$$\begin{pmatrix} 8 & 2 & 3 & 0 & 0 & 0 \\ 4 & 2 & 0 & 3 & 0 & 0 \\ 5 & 7 & 7 & 6 & 8 & 0 \\ 0 & 9 & 6 & 9 & 1 & 5 \\ 0 & 0 & 6 & 1 & 4 & 2 \\ 0 & 0 & 0 & 2 & 8 & 3 \end{pmatrix}$$

五对角矩阵

四、广义表

广义表（列表）： n (≥ 0)个表元素组成的有限序列，

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

LS是表名，a_i是表元素，它可以是表 (称为子表)，可以是数据元素(称为原子)。 n为表的长度。$n = 0$ 的广义表为空表。

1.广义表与线性表的区别？

- 线性表的成分都是结构上不可分的单元素；
- 广义表的成分可以是单元素，也可以是有结构的表；
- 线性表是一种特殊的广义表；
- 广义表不一定是线性表，也不一定是线性结构。

2.广义表的基本运算

求表头GetHead(L)

（1）

非空广义表的第一个元素，可以是一个单元素，也可以是一个子表。

求表尾GetTail(L)

（2）

非空广义表除去表头元素以外其它元素所构成的表。表尾一定是一个表。

A=( )	→	GetHead和GetTail均无定义
A=(a,b)	→	GetHead(A)=a GetTail(A)=(b)
A=(a)	→	GetHead(A)=a GetTail(A)=( )
A=((a))	→	GetHead(A)=(a) GetTail(A)=( )

A=(a,b,(c,d),(e,(f,g)))

GetHead(GetTail(GetHead(GetTail(GetTail(A))))) **d**

- 有次序性　一个直接前驱和一个直接后继
- 有长度　＝表中元素个数
- 有深度　＝表中括号的重数
- 可递归　自己可以作为自己的子表
- 可共享　可以为其他广义表所共享

小结

1. 明确数组和广义表这两种数据结构的特点;
2. 理解数组地址计算方法；
3. 了解几种特殊矩阵的压缩存储方法。