

# Lecture 12: 特殊矩阵与压缩存储

## 12.1 数组

### 12.1.1 一维数组

概念：一维存储

- 概念：
  - 各数组元素大小相同，且物理上连续存放。
  - 数组元素 `a[i]` 的存放地址 = `LOC + i * sizeof (ElemType)`
- 注意：
  - 除非题目特别说明，否则数组下标默认从 0 开始

数组的存储结构

```
//C语言定义一维数组
ElemType a[10];
```

### 12.1.2 二维数组

概念

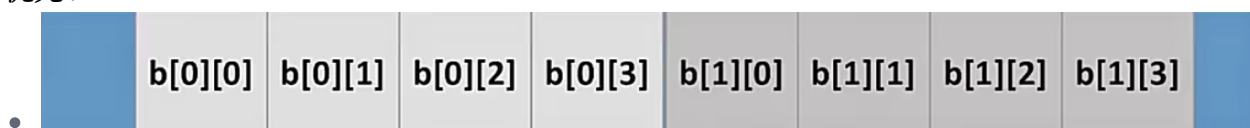
- 计算机内部从存储空间是线性的所以要是得逻辑——>物理
- 计算：
  - 行优先：`b[i][j]` 的存储地址 = `LOC + (i * N + j) * sizeof (ElemType)`
  - 列优先：`b[i][j]` 的存储地址 = `LOC + (j * M + i) * sizeof (ElemType)`

代码定义

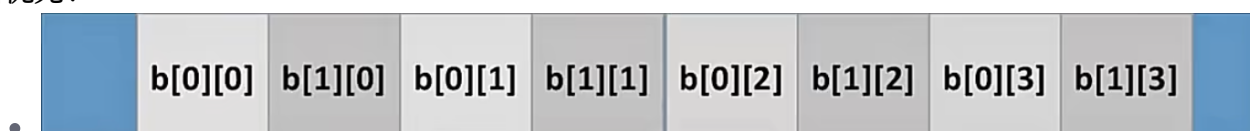
```
//2行4列的二维数组
ElemType b[2][4];
```

行优先与列优先

- 举例：2 行 4 列
- 行优先：



- 列优先：



- 图示：



起始地址: LOC

M行N列的二维数组  $b[M][N]$  中, 若按行优先存储, 则

- $b[i][j]$  的存储地址 =  $LOC + (i * N + j) * \text{sizeof}(\text{ElemType})$
- 列优先存储同上, 这里不做概述

## 12.2 特殊矩阵

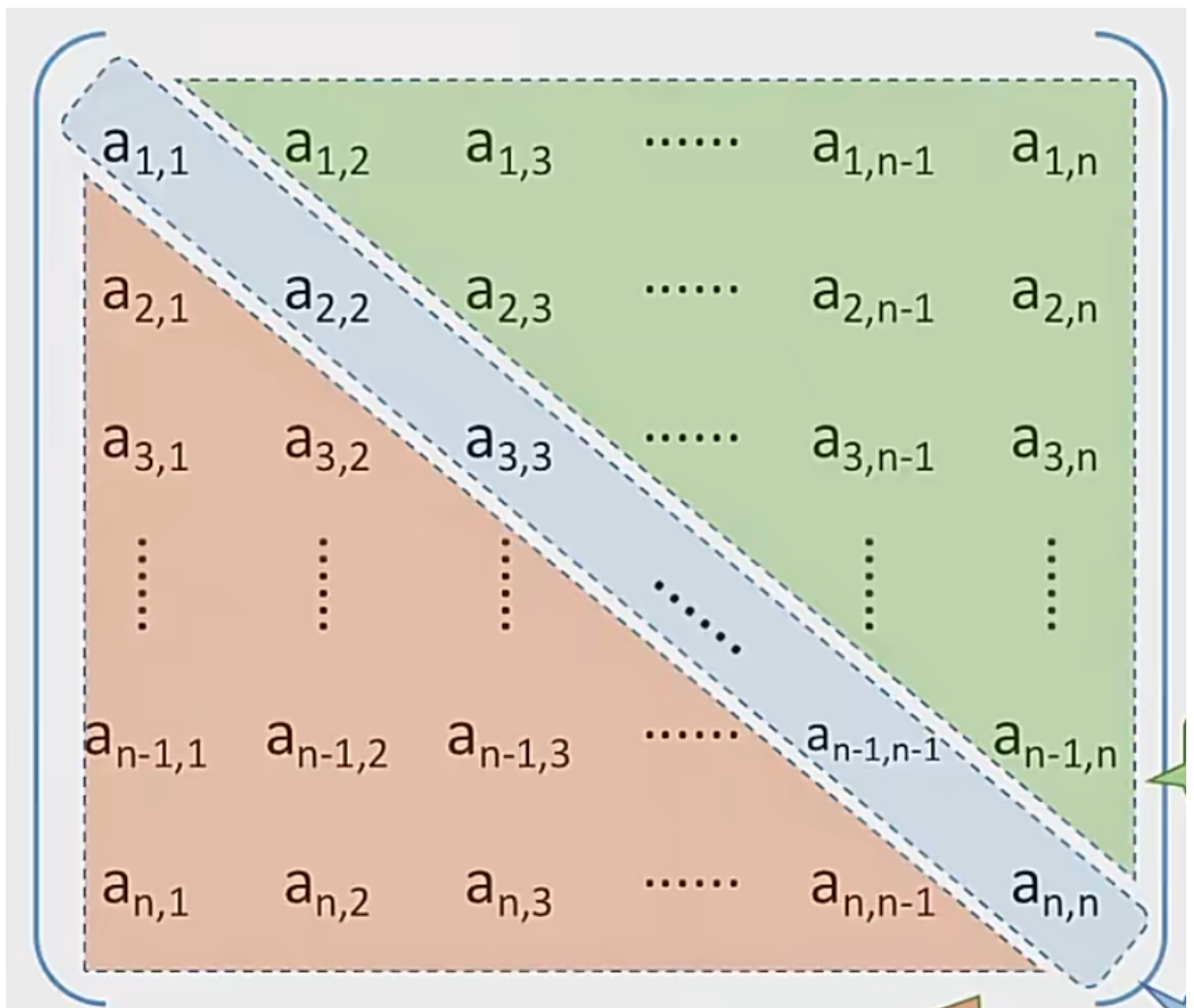
### 普通矩阵

- 可以直接使用二维数组存储;
- 注意 index 是从 0 开始还是 1 开始;

### 12.2.1 对称矩阵的压缩存储

**概念:** 对称矩阵的压缩存储

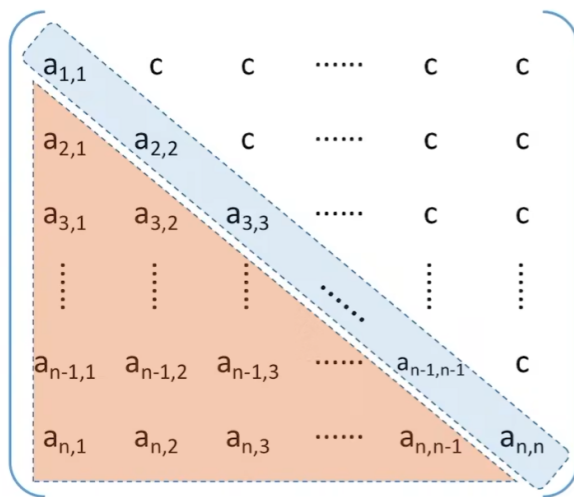
- 概念:
  - 若  $n$  阶方阵中任意一个元素  $a_{ij}$  都有  $a_{ij} = a_{ji}$  则该矩阵为对称矩阵;
- 普通存储:
  - $n * n$  二维数组
- 压缩存储策略:
  - 只存储主对角线+下三角区 (或主对角线+上三角区)
- 图示:



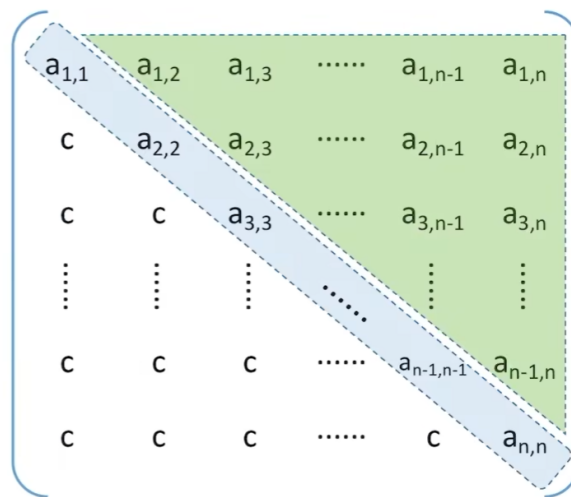
### 12.2.2 三对角矩阵的压缩存储

**概念：** 上三角矩阵与下三角矩阵

- 概念：
  - 当需要把对称矩阵中的数据进行存储时，可以只存储**主对角线+下三角区**；
  - 按行优先原则将各元素存入一维数组中；
  - 即：把对称矩阵压缩存储为一个三角矩阵；
- 概念：下三角的一维数组大小
  - $(1+n)*n/2$
- 图示：



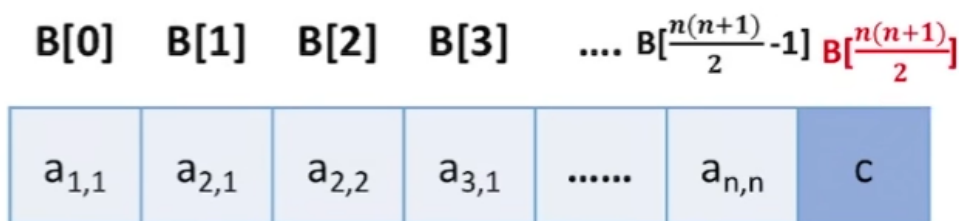
**下三角矩阵：**除了主对角线和下三角区，其余的元素都相同



**上三角矩阵：**除了主对角线和上三角区，其余的元素都相同

### 概念：压缩存储策略

- 概念：
  - 按行优先原则将橙色区元素存入**一维数组**中，并在最后一个位置存储常量 C；
- 图示：



矩阵下标  $\rightarrow$  一维数组下标

$a_{i,j} \ (i \geq j) \rightarrow B[k]$

- 公式：压缩存储 - 下三角

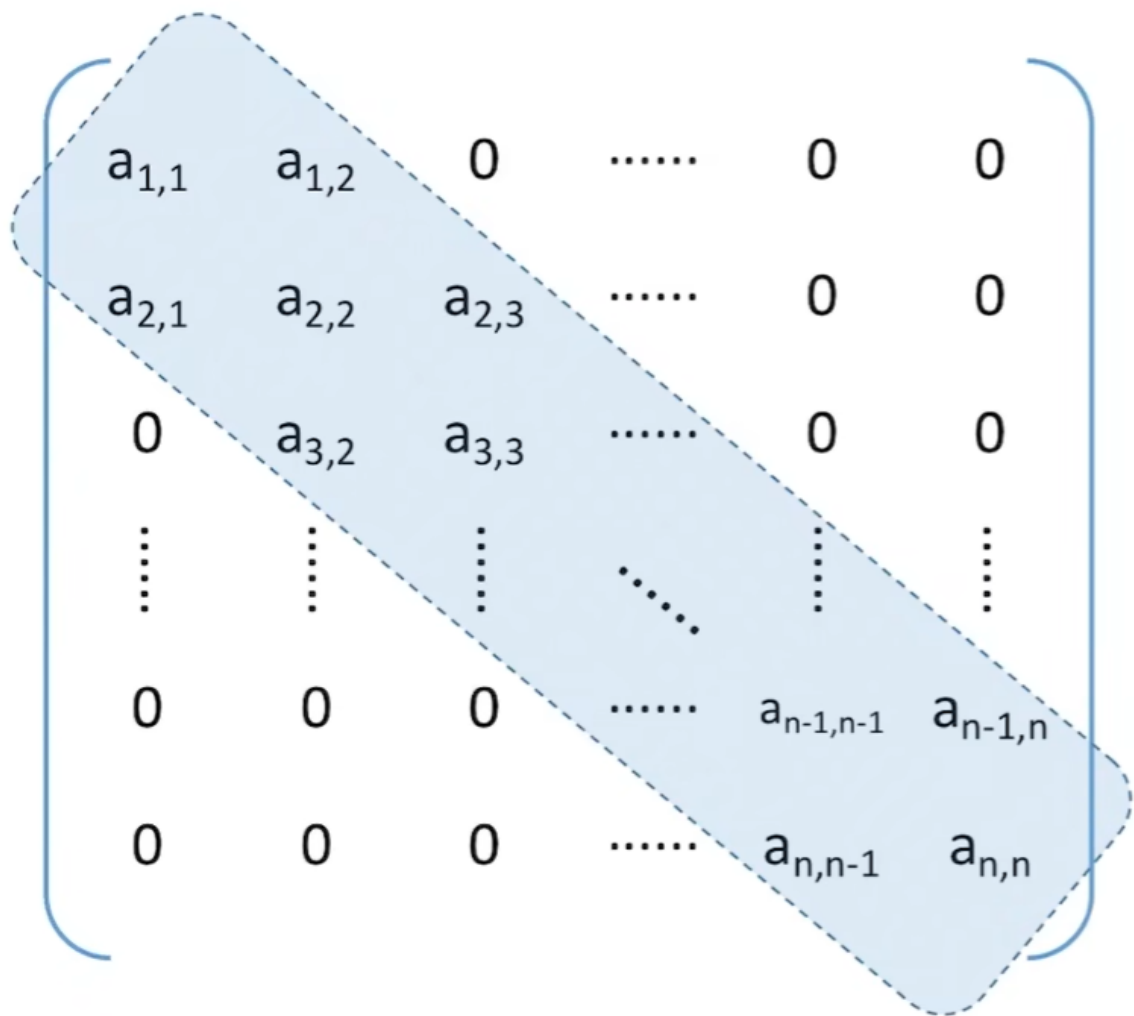
$$k = \begin{cases} \frac{i(i-1)}{2} + j - 1, & i \geq j \text{ (下三角区和对角线元素)} \\ \frac{n(n+1)}{2}, & i < j \text{ (上三角区元素)} \end{cases}$$

- 公式：压缩存储 - 上三角

$$k = \begin{cases} \frac{(i-1)(2n-i+2)}{2} + (j-i), & i \leq j \text{ (下三角区和对角线元素)} \\ \frac{n(n+1)}{2}, & i > j \text{ (上三角区元素)} \end{cases}$$

### 三对角矩阵的压缩存储

- 三对角矩阵，又称带状矩阵：当  $|i-j| > 1$  时，有  $a_{i,j} = 0 (1 \leq i, j \leq n)$
- 压缩存储策略：**
  - 按行优先(或列优先)原则，只存储带状部分；
- 图示：



- 注意：视频中是  $a_{ij}$  元素前有  $i-2$  个空(推导时注意)
- 压缩规则：  $a_{ij}(|i-j| \leq 1) \rightarrow B[k]$

### 12.2.3 稀疏矩阵的压缩存储

#### 稀疏矩阵的压缩存储

- **稀疏矩阵**：非零元素远远小于矩阵元素的个数
- **压缩存储策略**：
  - 顺序存储——三元组<行，列，值>
- 图示：

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 4 | 0 | 0 | 5 |
| 0 | 3 | 0 | 9 | 0 | 0 |
| 0 | 0 | 0 | 0 | 7 | 0 |
| 0 | 2 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

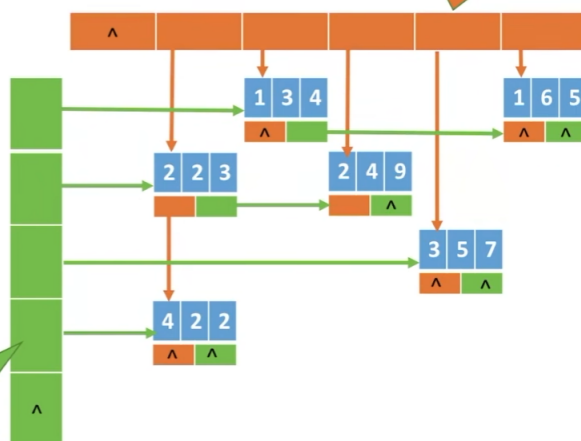
非零数据  
结点说明:

| 行              | 列              | 值 |
|----------------|----------------|---|
| 指向同列的<br>下一个元素 | 指向同行的<br>下一个元素 |   |

向右域 right,  
指向第 i 行的  
第一个元素

压缩存储策略二:  
链式存储——十字链表法

向下域 down,  
指向第 j 列的  
第一个元素



获取笔记持续更新可加微信: WandD2271