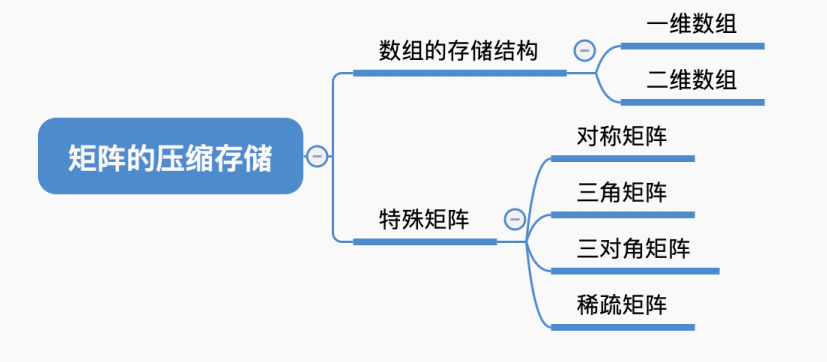
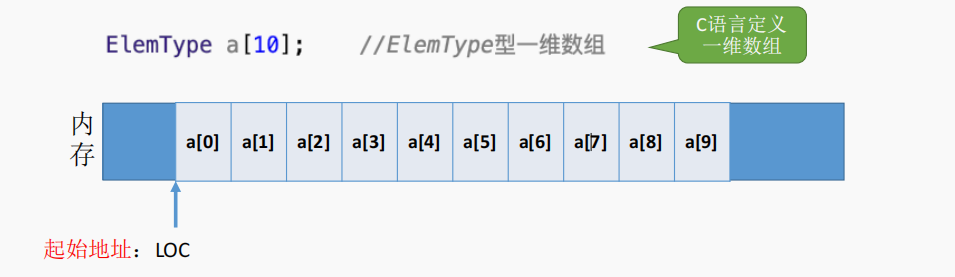
特殊矩阵压缩存储



一维数组的存储结构：



特点：

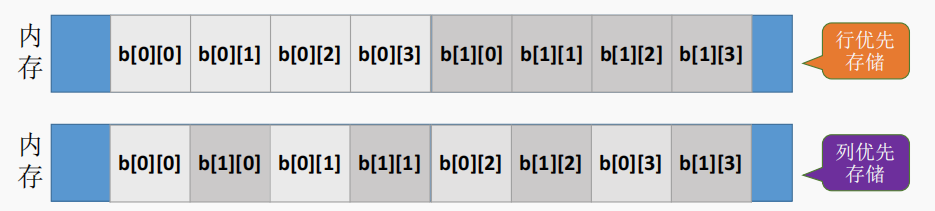
各数组**元素大小相同**，且**物理上连续存放**。

数组元素**a[i]的存放地址**=**LOC+i\*sizeof(ElemType)** (0≤i<10)

注：除非题目特别说明，否则数组下标默认**从0开始-**

二维数组的存储结构：

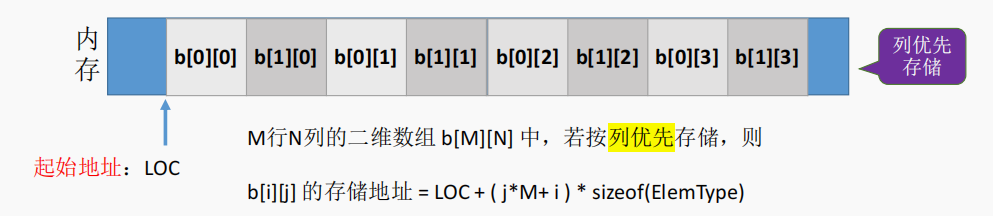






M行N列的二维数组b[M][N]中，若按**行优先存储**，

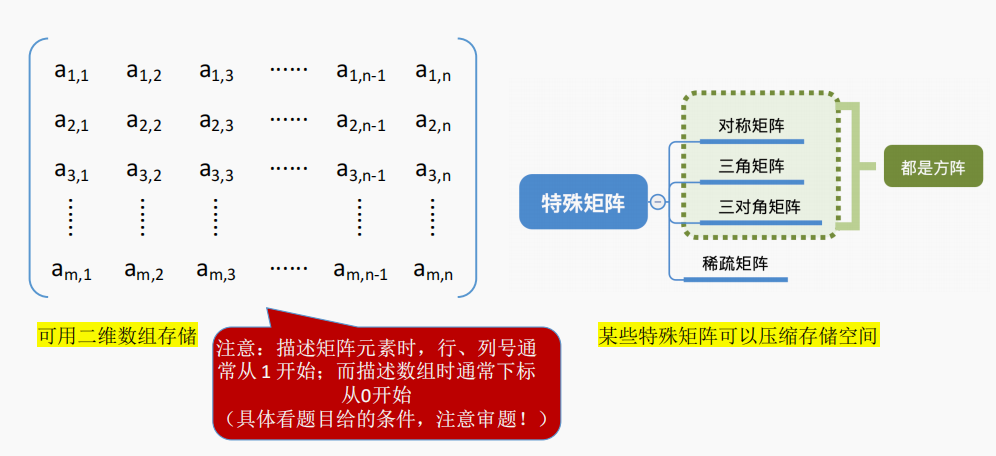
则**b[i][j]**的存储地址 = **LOC + (i\*N + j) \* sizeof(ElemType)**



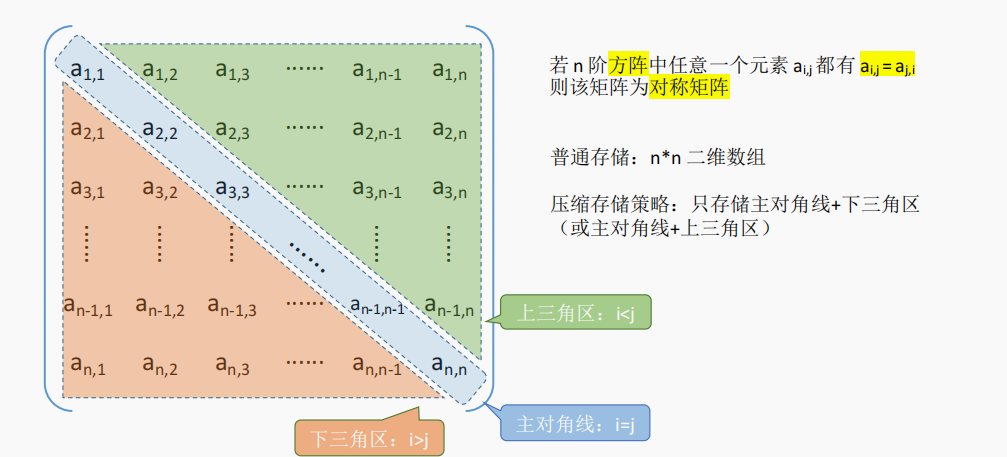
M行N列的二维数组b[M][N]中，若按**列优先存储**，

则**b[i][j]**的存储地址 = **LOC + ( j\*M+ i ) \* sizeof(ElemType)**

普通矩阵的存储（行列下标通常从1开始）：



对称矩阵的压缩存储：

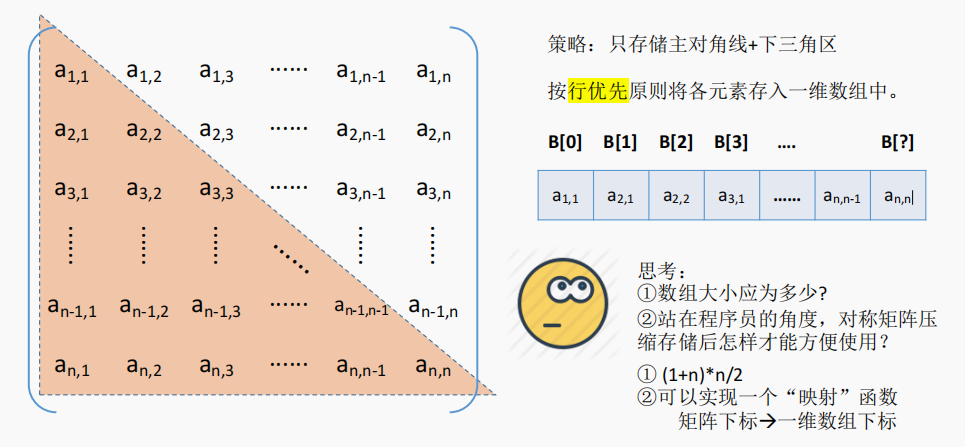


若 n 阶方阵中任意一个元素 ai,j都有**ai,j = aj,i**，则该矩阵为**对称矩阵**

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）

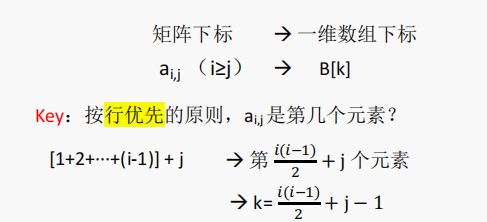
**按行优先**原则将各元素存入一维数组中。

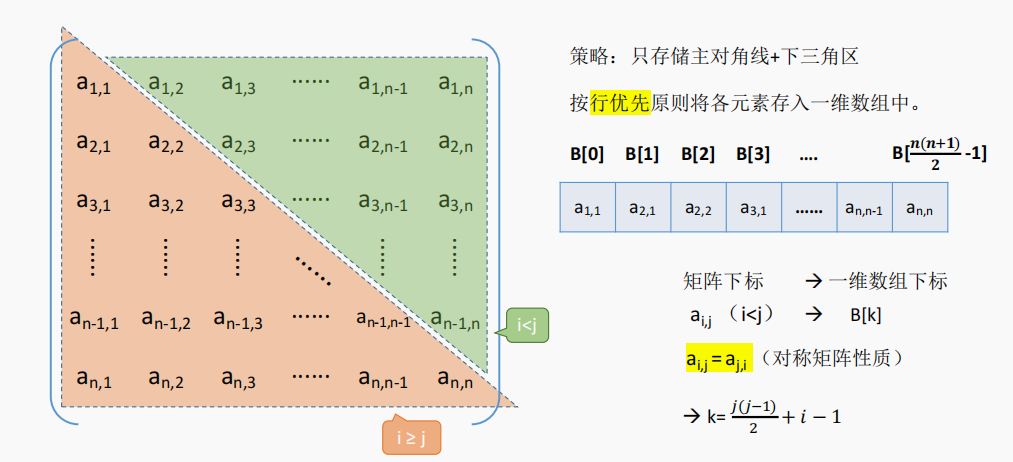


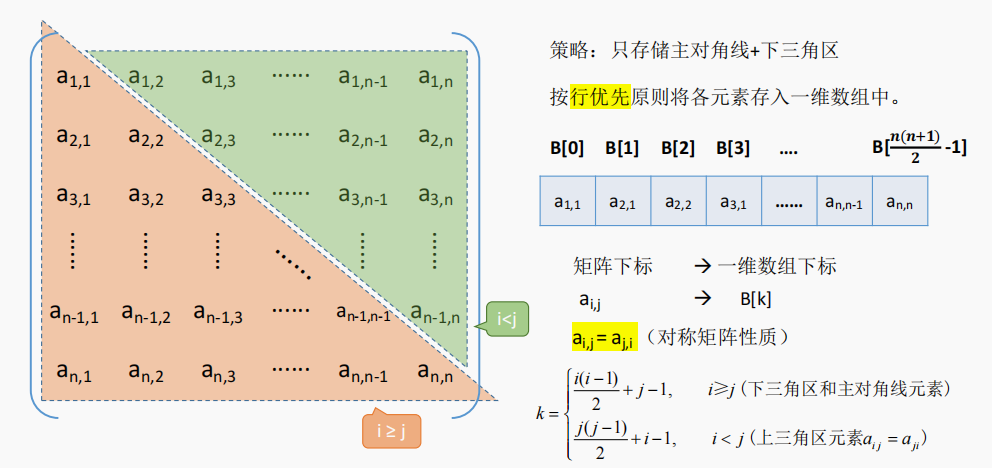
（1）数组大小应为**(1+n)\*n/2**

（2）站在程序员的角度，对称矩阵压缩存储后怎样才能方便使用？

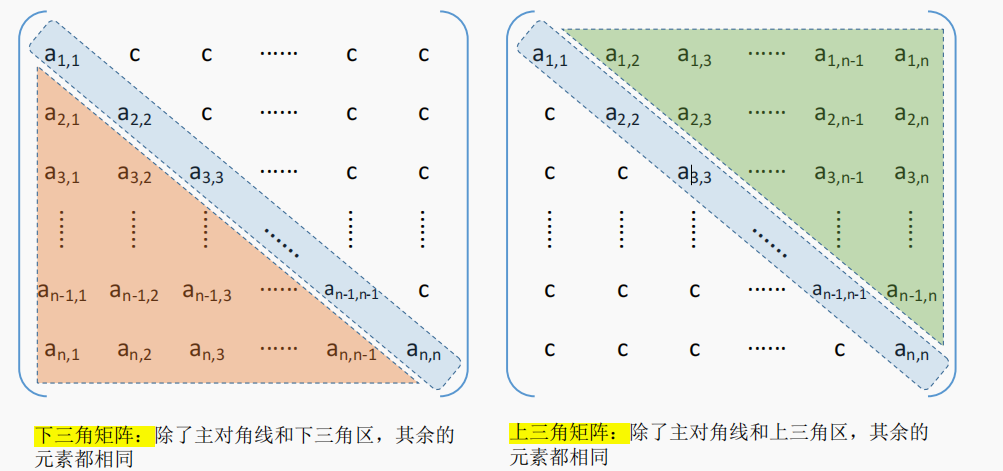
可以实现**一个“映射”函数矩阵下标**-->**一维数组下标**







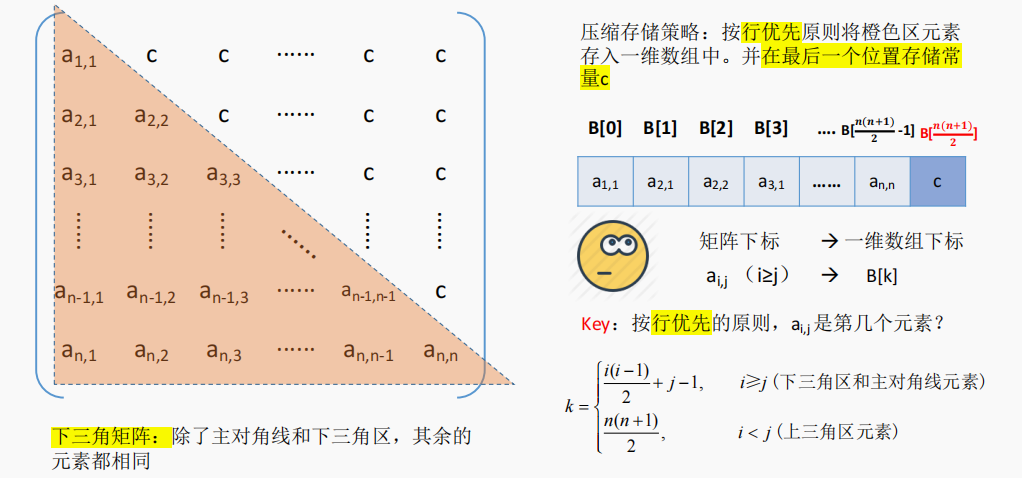
三角矩阵的压缩存储：

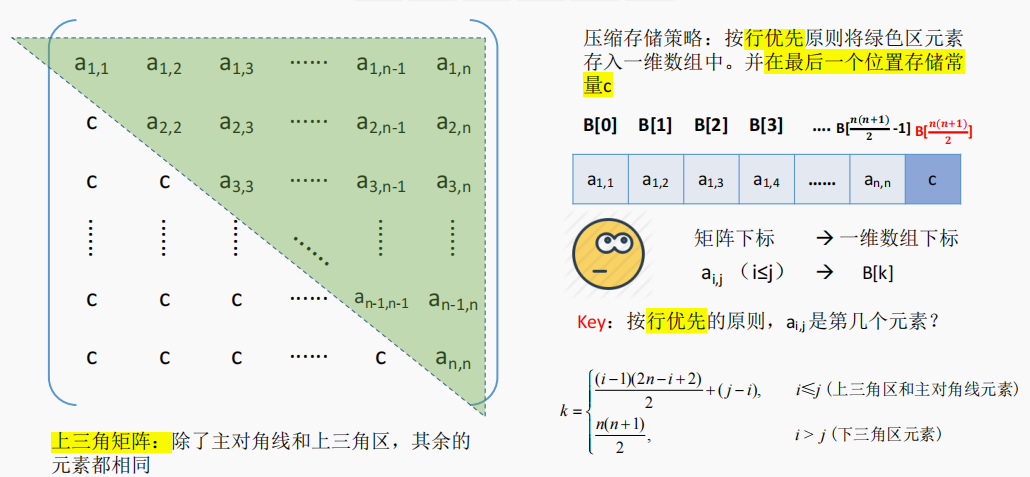


下三角矩阵：除了主对角线和下三角区，其余的元素都相同

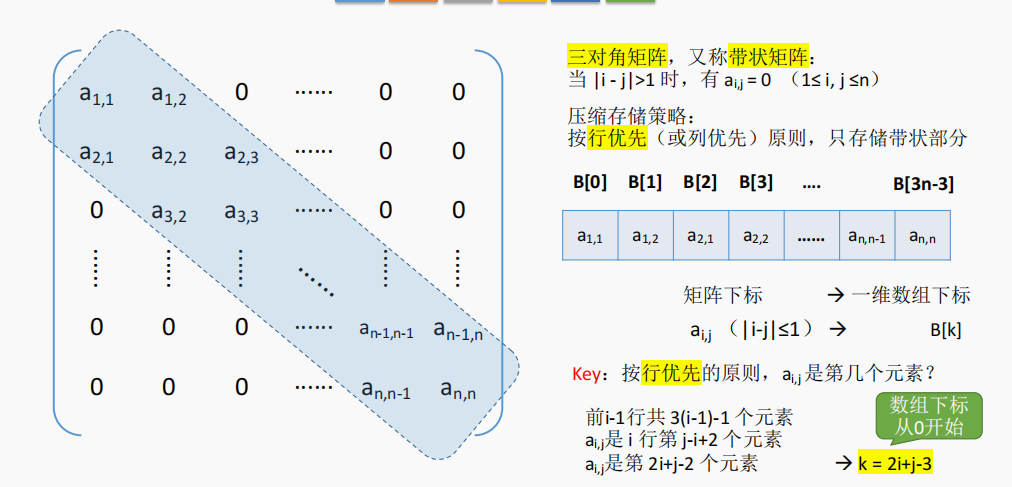
上三角矩阵：除了主对角线和上三角区，其余的元素都相同

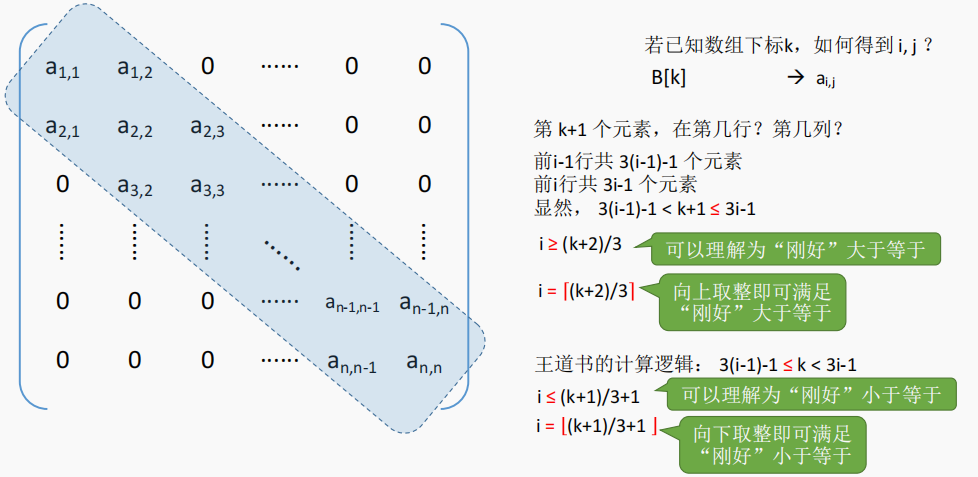
压缩存储策略：按**行优先**原则将**橙色区元素**存入**一维数组**中，并在最后一个位置**存储常量c**

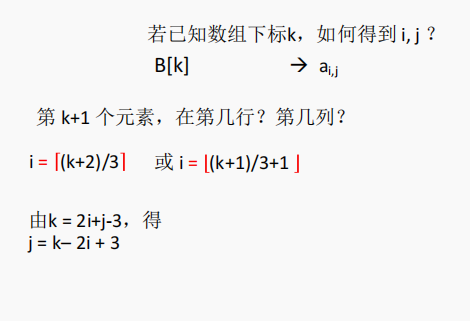




三对角矩阵的压缩存储：







稀疏矩阵的压缩存储：



