

数组和特殊矩阵

矩阵在计算机图形学、工程计算中占有举足轻重的地位。在数据结构中考虑的是如何用最小的内存空间来存储同样的一组数据。所以，我们不研究矩阵及其运算等，而把精力放在如何将矩阵更有效地存储在内存中，并能方便地提取矩阵中的元素。

数组的定义

数组是由 $n(n \geq 1)$ 个相同类型的数据元素构成的有限序列，每个数据元素称为一个数组元素，每个元素在 n 个线性关系中的序号称为该元素的下标，下标的取值范围称为数组的维界。

数组与线性表的关系：数组是线性表的推广。一维数组可视为一个线性表；二维数组可视为其元素是定长数组的线性表，以此类推。数组一旦被定义，其维数和维界就不再改变。因此，除结构的初始化和销毁外，数组只会有存取元素和修改元素的操作。

数组的存储结构

大多数计算机语言都提供了数组数据类型，逻辑意义上的数组可采用计算机语言中的数组数据类型进行存储，一个数组的所有元素在内存中占用一段连续的存储空间。

以一维数组 $A[0 \dots n-1]$ 为例，其存储结构关系式为

$$LOC(a_i) = LOC(a_0) + i \times L (0 \leq i < n)$$

其中， L 是每个数组元素所占的存储单元。

二维数组按行优先存储的下标对应关系

对于多维数组，有两种映射方法：按行有限或按列优先。以二维数组为例，按行优先存储的基本思想是：先行后列，先存储行号较小的元素，行号相等先存储列号较小的元素。设二维数组的行下标与列下标的范围分别为 $[0, h_1]$ 与 $[0, h_2]$ ，则存储结构关系式为

$$LOC(a_{i,j}) = LOC(a_{0,0}) + [i \times (h_2 + 1) + j] \times L$$

例如，对于数组 $A[2][3]$ ，它按行优先方式在内存中的存储形式如图所示。



图 3.18 二维数组按行优先顺序存放

当以列优先方式存储时，得出存储结构关系式为

$$LOC(a_{i,j}) = LOC(a_{0,0}) + [j \times (h_1 + 1) + i] \times L$$

例如，对于数组 $A[2][3]$ ，它按列优先方式在内存中的存储形式如图所示。

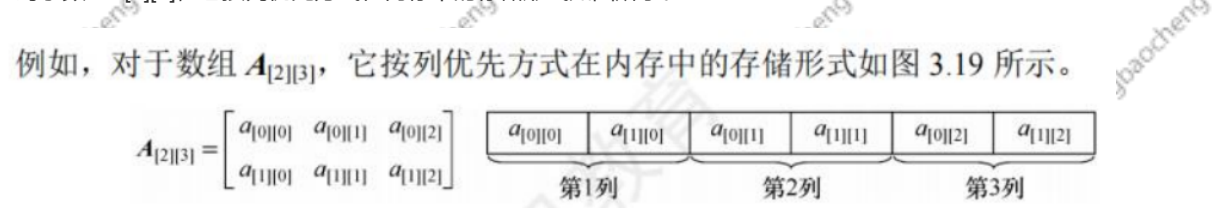


图 3.19 二维数组按列优先顺序存放

特殊矩阵的压缩存储

压缩存储	指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。
------	--------------------------------

特殊矩阵	指具有许多相同矩阵元素或零元素，并且这些相同矩阵元素或零元素的分布有一定规律性的矩阵。常见的特殊矩阵有对称矩阵、上（下）三角矩阵、对角矩阵等。
特殊矩阵的压缩存储方法	找出特殊矩阵中值相同的矩阵元素的分布规律，把那些呈现规律性分布的、值相同的多个矩阵元素压缩存储到一个存储空间中。

1. 对称矩阵

若对一个 n 阶矩阵 A 中的任意一个元素 $a_{i,j}$ 都有 $a_{i,j} = a_{j,i} (1 \leq i, j \leq n)$ ，则称其为对称矩阵。其中的元素可以划分为3个部分，即上三角区、主对角线和下三角区，如图所示。

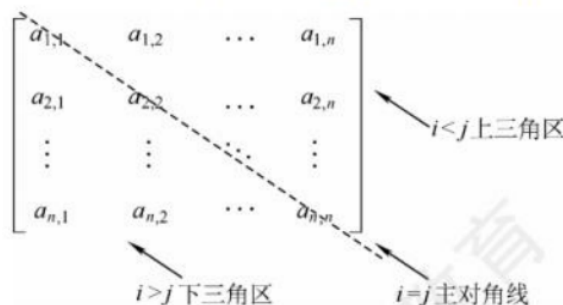


图 3.20 n 阶矩阵的划分

对于 n 阶对称矩阵，上三角区的所有元素和下三角区的对应元素相等，若仍采用二维数组存放，则会浪费几乎一半的时间

为此将 n 阶对称矩阵 A 存放在一维数组 $B[n(n+1)/2]$ 中，即元素 $a_{i,j}$ 存放在 b_k 中。比如只存放下三角部分(含主对角)的元素。

在数组 B 中，位于元素 $a_{i,j} (i \geq j)$ 前面的元素个数为

第1行：1个元素($a_{1,1}$)。

第2行：2个元素($a_{2,1}, a_{2,2}$)。

...

第 $i-1$ 行： $i-1$ 个元素($a_{i-1,1}, a_{i-1,2}, \dots, a_{i-1,i-1}$)。

第 i 行： $j-1$ 个元素($a_{i,1}, a_{i,2}, \dots, a_{i,j-1}$)。

因此，元素 $a_{i,j}$ 在数组 B 中的下标 $k = 1 + 2 + \dots + (i-1) + j - 1 = i(i-1)/2 + j - 1$ (数组下标从0开始)。

因此，元素下标之间的对应关系如下：

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

当数组下标从1开始时，可以采用同样的推导方法，请读者自行思考。

二维数组 $A[n][n]$ 和 $A[0 \dots n-1][0 \dots n-1]$ 的写法是等价的。若数组写为 $A[1 \dots n][1 \dots n]$ ，则说明指定了从下标1开始存储元素。二维数组元素写为 $a[i][j]$ ，注意数组元素下标 i 和 j 通常是从0开始的。矩阵元素通常写为 $a_{i,j}$ 或 $a_{(i)(j)}$ ，注意行号 i 和列号 j 是从1开始的。

2. 三角矩阵

下三角矩阵中，上三角区的所有元素均为同一常量。其存储思想与对称矩阵类似，不同之处在于存储完下三角区和主对角线上的元素之后，紧接着存储对角线上方的常量一次，所以可以将 n 阶下三角矩阵 A 压缩存储在 $B[n(n+1)/2+1]$ 中。

元素下标之间的对应关系为

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

下三角矩阵在内存中的压缩存储形式如图所示。

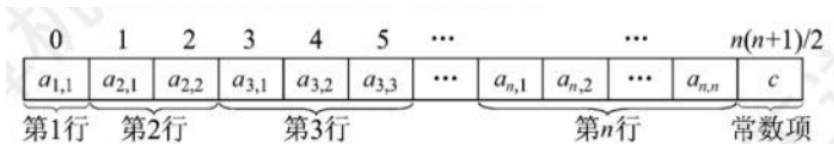


图 3.21 下三角矩阵的压缩存储

上三角矩阵采用行优先存储的应用

上三角矩阵中，下三角区的所有元素均为同一常量。只需存储主对角线、上三角区上的元素和下三角区的常量一次，可将其压缩存储在 $B[n(n+1)/2+1]$ 中。

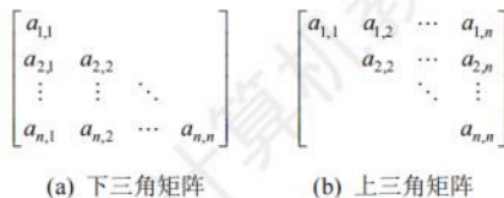


图 3.22 三角矩阵

在数组 B 中，位于元素 $a_{i,j}$ ($i \leq j$) 前面的元素个数为

第1行： n 个元素

第2行： $n-1$ 个元素

...

第 $i-1$ 行： $n-i+2$ 个元素

第 i 行： $j-i$ 个元素

因此，元素 $a_{i,j}$ 在数组 B 中的下标

$$k = n + (n-1) + \dots + (n-i+2) + (j-i+1) - 1 = (i-1)(2n-i+2)/2 + (j-i)$$

因此，元素下标之间的对应关系如下：

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

上三角矩阵在内存中的压缩存储形式如图所示。

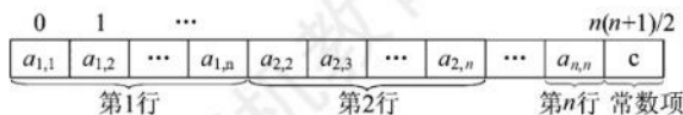


图 3.23 上三角矩阵的压缩存储

以上推到均假设数组的下标从0开始，若题设有具体要求，则应该灵活应对。

3. 三对角矩阵

对角矩阵也称带状矩阵。对 n 阶矩阵 A 中的任意一个元素 $a_{i,j}$ ，当 $|i-j| > 1$ 时，若有 $a_{i,j} = 0$ ($1 \leq i, j \leq n$)，则称三对角矩阵，如图所示。在三对角矩阵中，所有非零元素都集中在以主对角线为中心的3条对角线的区域，其他区域的元素都为零。

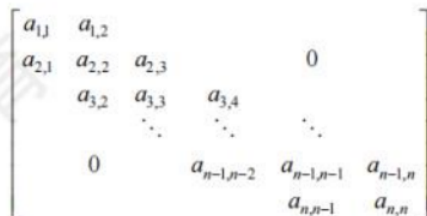


图 3.24 三对角矩阵 A

三对角矩阵 A 也可以采用压缩存储，将3条对角线上的元素按行优先方式存放在一维数组 B 中，且 $a_{1,1}$ 存放在 $B[0]$ 中，其存储形式如图所示。

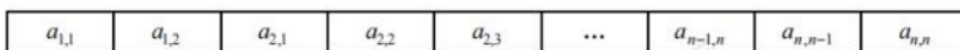


图 3.25 三对角矩阵的压缩存储

三对角矩阵压缩存储的下标对应关系

由此可以计算矩阵 A 中3条对角线上的元素 $a_{i,j}$ ($1 \leq i, j \leq n, |i-j| \leq 1$)在一维数组 B 中存放的下标为 $k = 2i + j - 3$.

反之,若已知三对角矩阵中的某个元素 $a_{i,j}$ 存放在一维数组 B 的第 k 个位置,则有 $i = \lfloor (k+1)/3 + 1 \rfloor$, $j = k - 2i + 3$.例如,当 $k = 0$ 时, $i = \lfloor (0+1)/3 + 1 \rfloor = 1$, $j = 0 - 2 \times 1 + 3 = 1$,存放的是 $a_{1,1}$;当 $k = 2$ 时, $i = \lfloor (2+1)/3 + 1 \rfloor = 2$, $j = 2 - 2 \times 2 + 3 = 1$,存放的是 $a_{2,1}$;当 $k = 4$ 时, $i = \lfloor (4+1)/3 + 1 \rfloor = 2$, $j = 4 - 2 \times 2 + 3 = 3$,存放的是 $a_{2,3}$.

稀疏矩阵

矩阵中非零元素的个数 t ,相对矩阵元素的个数 s 来说非常少,即 $s \gg t$ 的矩阵称为稀疏矩阵.

例如,一个矩阵的阶为 100×100 ,该矩阵中只有少于100个非零元素.

存储系数矩阵需要保存的信息 (2023)

若采用常规的方法存储稀疏矩阵,则相当浪费存储空间,因此仅存储非零元素.但通常非零元素的分布没有规律,所以仅存储非零元素的值是不够的,还要存储它所在的行和列.

因此,将非零元素及其相应的行和列构成一个三元组(行标 i ,列标 j ,值 $a_{i,j}$),如图所示.

然后按照某种规律存储这些三元组线性表.稀疏矩阵压缩存储后便失去了随机存取特性.

$M = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 23 & 0 & 0 \end{bmatrix}$	对应的三元组	<table><tr><th>i</th><th>j</th><th>$a_{i,j}$</th></tr><tr><td>0</td><td>0</td><td>4</td></tr><tr><td>1</td><td>2</td><td>6</td></tr><tr><td>2</td><td>1</td><td>9</td></tr><tr><td>3</td><td>1</td><td>23</td></tr></table>	i	j	$a_{i,j}$	0	0	4	1	2	6	2	1	9	3	1	23
i	j	$a_{i,j}$															
0	0	4															
1	2	6															
2	1	9															
3	1	23															

图 3.26 稀疏矩阵及其对应的三元组

适合系数矩阵压缩存储的存储结构 (2017)

系数矩阵的三元组表既可以采用数组存储,又可以采用十字链表存储.当存储稀疏矩阵时,不仅要保存三元组表,而且要保存系数矩阵的行数、列数和非零元素的个数.