第6章 数组和稀疏矩阵

6.1 数　组

6.2 稀疏矩阵

6.1 数　组

6.1.1 数组的基本概念

从逻辑结构上看，一维数组A是n（$n>1$）个相同类型数据元素a_1、a_2、…、a_n构成的有限序列，其逻辑表示为：

$$A=(a_1, a_2, \ldots, a_n)$$

其中，a_i（$1 \leqslant i \leqslant n$）表示数组$A$的第$i$个元素。

一个m行n列的二维数组A可以看作是每个数据元素都是相同类型的一维数组的一维数组。

$$A=\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \cdots & & & \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \Rightarrow A=[\,A_1,\ A_2,\ \ldots,\ A_m\,]$$

$$A_1=[\,a_{1,1},\ a_{1,2},\ \ldots,\ a_{1,n}\,]$$

$$A_2=[\,a_{1,1},\ a_{2,2},\ \ldots,\ a_{2,n}\,]$$

$$\cdots\ \ \cdots$$

$$A_m=[\,a_{m,1},\ a_{m,2},\ \ldots,\ a_{m,n}\,]$$

由此看出，多维数组是线性表的推广。

数组抽象数据类型＝逻辑结构＋基本运算（运算描述）

数组的基本运算如下：

❶ Value(A,index$_1$,index$_2$,…,index$_d$)：即A(index$_1$,index$_2$,…,index$_d$)=e，元素赋值。

❷ Assign(A,e,index$_1$,index$_2$,…,index$_d$)：即e= A(index$_1$,index$_2$,…,index$_d$)，取元素值。

❸ ADisp(A,b_1,b_2,…,b_d)：输出d维数组A的所有元素值。

6.1.2 数组的存储结构

将数组的所有元素存储在一块地址连续的内存单元中，这是一种顺序存储结构。

几乎所有的计算机语言都支持数组类型，以C/C++语言为例，其中数组数据类型具有以下性质：

- 数组中的数据元素数目固定。
- 数组中的所有数据元素具有相同的数据类型。
- 数组中的每个数据元素都有一组唯一的下标。
- 数组是一种随机存储结构。可随机存取数组中的任意数据元素。

一维数组：一旦a_1的存储地址LOC(a_1)确定，并假设每个数据元素占用k个存储单元，则任一数据元素a_i的存储地址LOC(a_i)就可由以下公式求出：

$$\mathrm{LOC}(a_i)=\mathrm{LOC}(a_1)+(i-1)*k \qquad (0\leqslant i\leqslant n)$$

数组a：$\overbrace{a_1\quad a_2\quad a_3\quad \ldots\quad a_{i-1}}^{\text{共}i-1\text{个元素}}\quad a_i\quad \ldots\quad a_n$

表明一维数组具有随机存储特性。

对于一个m行n列的二维数组$A_{m\times n}$，存储方式：

- 以行序为主序的存储
- 以列序为主序的存储

❶ 以行序为主序的存储方式

$$A=\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \cdots & & & \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

$$\underbrace{a_{1,1},\ a_{1,2},\ \ldots,\ a_{1,n}}_{\text{第1行的元素}}\quad \underbrace{a_{2,1},\ a_{2,2},\ \ldots,\ a_{2,n}}_{\text{第2行的元素}}\quad \ldots,\quad \underbrace{a_{m,1},\ a_{m,2},\ \ldots,\ a_{m,n}}_{\text{第}m\text{行的元素}}$$

第1行的元素　　　　第i行的元素

$$a_{1,1},\ a_{1,2},\ \dots,\ a_{1,n},\ \dots,\ a_{i,1},\ a_{i,2},\ \dots,\ a_{i,j-1},\ a_{i,j},\ \dots a_{i,n},\ \dots$$

1~i−1行，每行n个元素，
计$(i-1) \times n$个元素

第i行中，$a_{i,j}$元素前
有$j-1$个元素

则$a_{i,j}$元素前共有 $(i-1)\times n+j-1$个元素

$$LOC(a_{i,j})=LOC(a_{1,1})+[(i-1)\times n+(j-1)] \times k$$

❷ 以列序为主序的存储方式

同理可推出在以列序为主序的计算机系统中有：

$$\mathrm{LOC}(a_{i,j})=\mathrm{LOC}(a_{1,1})+[(j-1)\times m+(i-1)]\times k$$

其中m为行数。

所以，二维数组采用顺序存储结构时，也具有随机存取特性。

是指给定序号i（下标），可以在O(1)的时间内找到相应的元素值。

同样，多维数组采用顺序存储时具有随机存储特性。

6.1.3 特殊矩阵的压缩存储

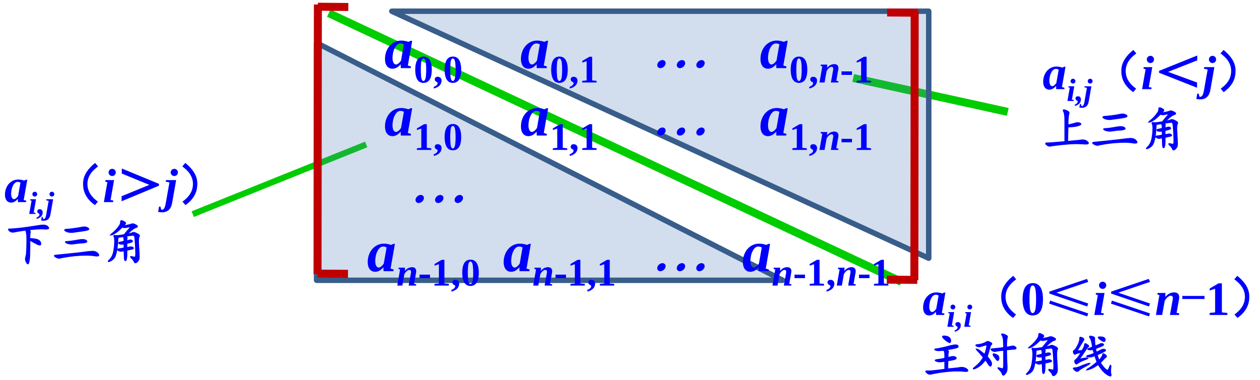
特殊矩阵的主要形式有：

- 对称矩阵
- 上三角矩阵 / 下三角矩阵
- 对角矩阵

它们都是方阵，即行数和列数相同。

1、对称矩阵的压缩存储

若一个n阶方阵$A[n][n]$中的元素满足$a_{i,j}=a_{j,i}$（$0\leqslant i$，$j\leqslant n-1$），则称其为n阶对称矩阵。



以行序为主序存储其下三角+主对角线的元素。

$$\begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n-1} \\ \cdots & & & \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-1} \end{bmatrix}$$

$i \geqslant j$

下三角+主对角线

$a_{i,j} \to b_k$　　$k = ?$

$$a_{0,0},\quad a_{1,0},\quad a_{1,1},\quad \cdots,\quad a_{n-1,0},\quad a_{n-1,1},\quad \cdots,\quad a_{n-1,n-1}$$

$$B=\ (\underbrace{b_0,\quad b_1,\quad b_2,\quad \ldots\ldots,\quad \ldots\ldots,\quad b_s}_{n(n+1)/2\text{个元素}})$$

$$b_k \updownarrow$$

$$B=(\underbrace{a_{0,0}}_{1\text{个元素}},\underbrace{a_{1,0},a_{1,1}}_{2\text{个元素}},\ldots,\underbrace{a_{i-1,0},\ \ldots\ ,a_{i-1,i-1}}_{i\text{个元素}},\ \underbrace{a_{i,0}\ ,\ \ldots,\ a_{i,j-1}}_{j\text{个元素}},\ a_{i,j},\ \ldots,\ a_{n-1,n-1})$$

共计$i(i+1)/2+j$个元素

$$k=\begin{cases}\dfrac{i(i+1)}{2}+j & \text{当 } i\geq j\text{时（下三角+主对角线的元素）}\\[2ex] \dfrac{j(j+1)}{2}+i & \text{当} i<j\text{时（}a_{i,j}=a_{j,i}\text{）}\end{cases}$$

n^2个元素 $\longleftrightarrow$ $n(n+1)/2$个元素

‖ $\longleftrightarrow$ ‖

$A[0..n-1,0..n-1]$ $\quad\quad$ $B[0..n(n+1)/2-1]$

$\downarrow$ $\quad\quad$ $\downarrow$

$a[i][j]$ $\longleftrightarrow$ $b[k]$

$\Downarrow$

$$k=\begin{cases} \dfrac{i(i+1)}{2}+j & \text{当 } i\geqslant j\text{时} \\ \dfrac{j(j+1)}{2}+i & \text{当} i<j\text{时}(a_{i,j}=a_{j,i}) \end{cases}$$

对于对称矩阵A，采用一维数组B存储，并提供A的所有运算。

2、 三角矩阵的压缩存储

● 上三角矩阵：

$$\begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ & a_{1,1} & \cdots & a_{1,n-1} \\ c & & & \cdots \\ & & & a_{n-1,n-1} \end{bmatrix} \quad i \leqslant j$$

$$a_{i,j} \rightarrow b_k$$

$$B=\underbrace{(\underbrace{a_{0,0},a_{0,1},\ldots,a_{0,n-1}}_{n\text{个元素}},\underbrace{a_{1,1},\ldots,a_{i-1,n-1}}_{n-1\text{个元素}},\ldots,\underbrace{a_{i-1,i-1},\ldots,a_{i-1,n-1}}_{n-i+1\text{个元素}},\underbrace{a_{i,i},\ldots,a_{i,j-1}}_{j-i\text{个元素}}}_{\text{共计}i(2n-i+1)/2+j-i\text{个元素}},a_{i,j},\ldots)$$

共计$i(2n-i+1)/2+j-i$个元素

$$k=\begin{cases} \dfrac{i(2n-i+1)}{2}+j-i & \text{当}i\leqslant j\text{时} \\ \dfrac{n(n+1)}{2} & \text{当}i>j\text{时} \end{cases}$$

存放常量c

- 下三角矩阵

$$i \geqslant j \qquad \begin{bmatrix} a_{0,0} & & c & \\ a_{1,0} & a_{1,1} & & \\ \cdots & & & \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-1} \end{bmatrix} \qquad a_{i,j} \rightarrow b_k$$

$$B=(a_{0,0},\ a_{1,0},\ a_{1,1},\ \ldots,\ a_{n-1,0},\ a_{n-1,1},\ \ldots,\ a_{n-1,n-1})$$

$$k=\begin{cases} \dfrac{i(i+1)}{2}+j & \text{当 } i \geqslant j \text{时} \\ \dfrac{n(n+1)}{2} & \text{当 } i<j \text{时} \end{cases}$$

存放一个常量c

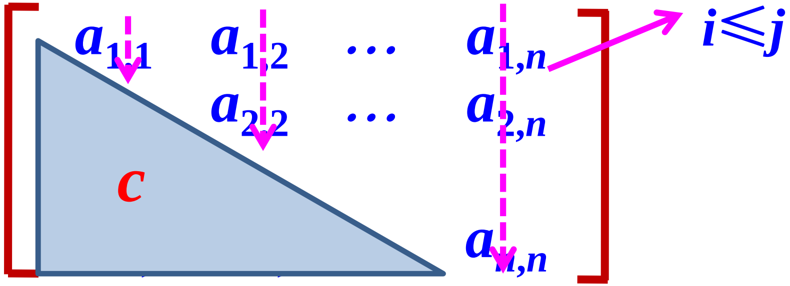
【例6-1】若将n阶上三角矩阵A按列优先顺序压缩存放在一维数组$B[1..n(n+1)/2]$中，A中第一个非零元素$a_{1,1}$存于B数组的b_1中，则应存放到b_k中的非零元素$a_{i,j}$（$i \leqslant j$）的下标i、j与k的对应关系是________。

A. $i(i+1)/2+j$

B. $i(i-1)/2+j$

C. $j(j+1)/2+i$

D. $j(j-1)/2+i$



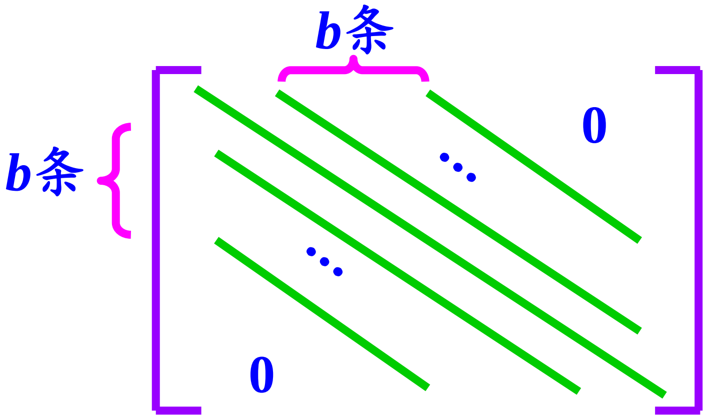
$1 \sim j-1$列的元素个数：$j(j-1)/2$

第j列a_{ij}之前的元素个数：$i-1$

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

- 按行还是按列
- 初始下标从0还是从1开始

3、对角矩阵的压缩存储



半带宽为b的对角矩阵

对角矩阵　　　　压缩存储

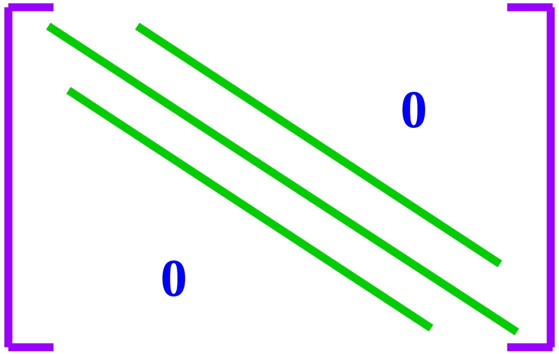
$A \longleftrightarrow B$

$a[i][j] \longleftrightarrow b[k]$

当b=1时称为三对角矩阵

其压缩地址计算公式如下：

$$k = 2i + j$$



思考题：

特殊矩阵为什么采用压缩存储，需要解决什么问题？

——本讲完——