jyang
@cs.ecnu.edu.cn

第四章 数组

1 数组的顺序存储

2 稀疏矩阵

jyang
@cs.ecnu.edu.cn

1 数组的顺序存储

数组是线性表的扩展：表中的数据元素本身也是一个数据结构。

所谓数组的顺序存储，就是把数组中各个元素的值按一定的次序存放在计算机的一组连续存储单元中。

数组顺序存储的优点：可以随机存取或修改数组元素的值，只要知道数组元素的下标值，就可以按相应的**地址计算公式**求得该元素的存放地址。

jyang
@cs.ecnu.edu.cn

1 数组的顺序存储

一维数组和二维数组

对于C语言，数组a[t1]是由元素a[0],a[1],…，a[t1-1]组成的有限序列。

如果数组中的类型为datatype，那么令 s=sizeof(datatype)，它是一个数组元素所占用的存储单元个数。

一维数组的地址计算公式： &a[i]=&a[0]+i*S 0<i<=t1-1

jyang
@cs.ecnu.edu.cn

1 数组的顺序存储

一维数组和二维数组

对于C语言的二维数组a[t1][t2]，可以写成t1*t2的矩阵形式。

$a_{0\,0}$ $a_{0\,1}$ …… $a_{0\,t2-1}$

$a_{1\,0}$ $a_{1\,1}$ …… $a_{1\,t2-1}$

……

$a_{t1-1\,0}$ $a_{t1-1\,1}$ …… $a_{t1-1\,t2-1}$

jyang
@cs.ecnu.edu.cn

1 数组的顺序存储

一维数组和二维数组

对于C语言的二维数组a[t1][t2]，顺序分配一般有两种常用的方式：一种是按行序优先，一种是列序优先。

行序优先：先存行号为0的元素，然后存行号为1的元素……
同行号元素，列号小的先存。

地址：

$\&a[i][0]=\&a[0][0]+i*t_2*S$

$\&a[i][j]=\&a[i][0]+j*S$

$\&a[i][j]=\&a[0][0]+i*t_2*S+j*S=\&a[0][0]+(i*t_2+j)*S$

jyang
@cs.ecnu.edu.cn

1 数组的顺序存储

一维数组和二维数组

对于C语言的二维数组a[t1][t2]，顺序分配一般有两种常用的方式：一种是按行序优先，一种是列序优先。

列序优先：先存列号为0的元素，然后存列号为1的元素……
同列号元素，行号小的先存。

地址：

&a[0][j]=&a[0][0]+j*t_1*S

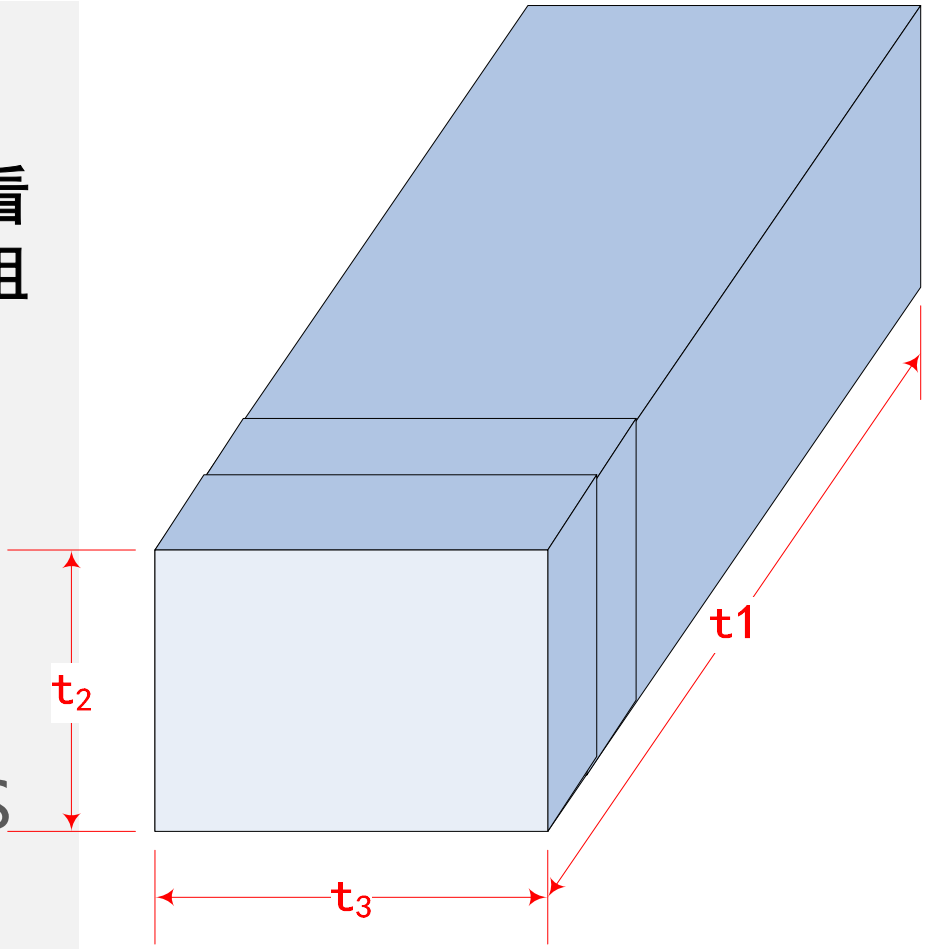
&a[i][j]=&a[0][j]+i*S

&a[i][j]=&a[0][0]+(j*t_1+i)*S

jyang
@cs.ecnu.edu.cn

三维数组和n维数组

三维数组a[t1][t2][t3]可以看成由t1个阶为t2*t3的二维数组所组成。



按行序:

&a[i][j][k]=&a[0][0][0]
+（i*t2*t3+j*t3+k）*S

三维数组和n维数组

n维数组a[t1][t2]……[tn]的地址计算公式。

$\&a[i_1][i_2]...[i_n]=\&a[0][0]...[0]+S\times\sum_{j=1}^{n} i_j * C_j$

在此处键入公式。

其中，$C_j=\prod_{k=j+1}^{n} t_k$　　(1≤j < n)

$C_n=1$

$C_j=t_{j+1}*C_{j+1}$

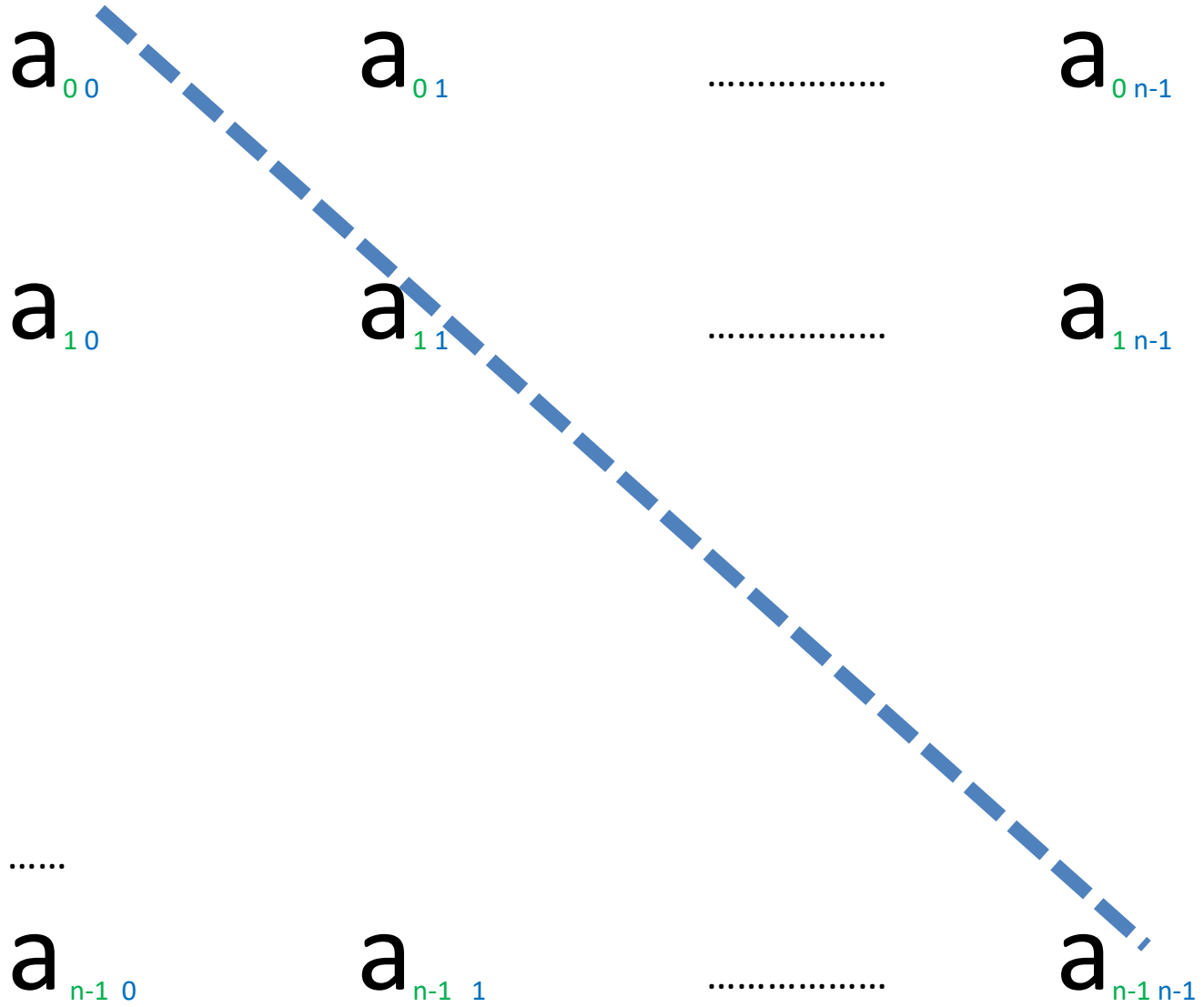
三角矩阵和带状矩阵

下三角矩阵：上三角（不包括最长主对角线）中的元素都是零的n阶方阵。
上三角矩阵：……
带状矩阵：对于n阶方阵，如果我们只考虑以最长主对角线为中心的带状区域中的元素，而其他元素都为零，这个带状区域包含最长主对角线下面及上面各b条主对角线上的元素，那么我们称这样的方阵为半带宽为b的带状矩阵，或带宽为（2b+1）的带状矩阵。

如何用数组存放这些特殊的矩阵？

jyang
@cs.ecnu.edu.cn

第四章 数组



jyang
@cs.ecnu.edu.cn

1 数组的顺序存储

三角矩阵和带状矩阵

下三角矩阵：N阶下三角，非零元素的下标满足如下规律，

$$0 \le j \le i \le n-1$$

如果按照行序优先进行顺序存储，不存非零元素，

$a_{00},a_{10},a_{11},a_{20},a_{21},a_{22},a_{30},\cdots\cdots\cdots a_{n-1\ n-1}$

$\&a[i][0]=\&a[0][0]+s*\sum_{k=1}^{i}k=\ \&a[0][0]+s\times\frac{i(i+1)}{2}$

$\&a[i][j]=\&a[i][0]+j*s$

$\&a[i][j]=\&a[0][0]+\left[\frac{i(i+1)}{2}+j\right]*s$

三角矩阵和带状矩阵

N阶上三角矩阵？

jyang
@cs.ecnu.edu.cn

1 数组的顺序存储

三角矩阵和带状矩阵

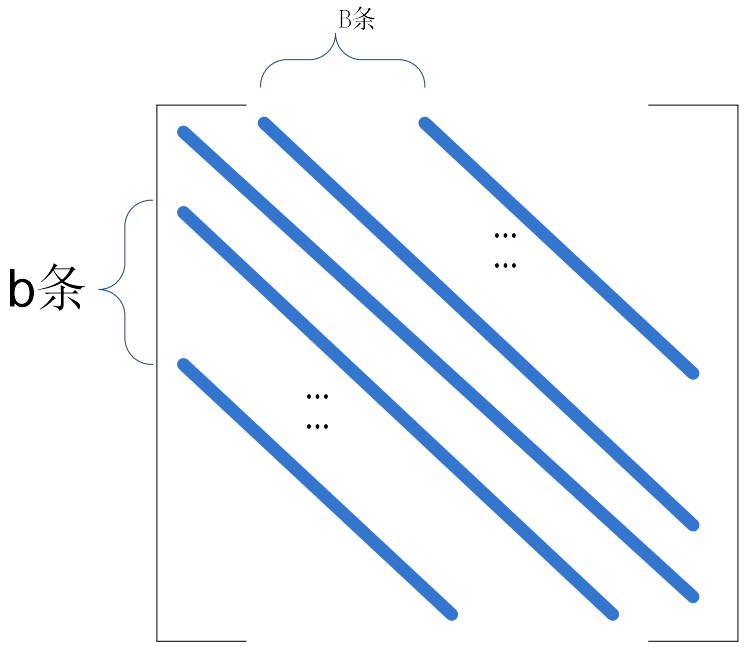
带宽：2b+1

值为0的元素：半带宽为b（0<=b<n）时，当0<=i，j<=n-1且 |i-j|>b时，a_{ij}=0。

存储：不存储满足|i-j|>b的元素。

需要存储的元素的个数：
(2b+1)n-b(b+1)

值为0的元素下标有什么特点？
一共多少个非零元素？



半带宽为b的带状矩阵

三角矩阵和带状矩阵

地址公式：为了使地址计算公式简单，可以使用如下的存储方法：除头一行和最后一行外，每行都当作有（2b+1）个元素。将带状矩阵按行序优先存储在（2b+1）n-2b个存储单元中。

如果按照行序列序存放带宽矩阵，

&a[i][i]=&a[i-1][i-1] +(2b+1)*S

&a[i][i]=&a[0][0] +i*(2b+1)*S

&a[i][j]=&a[i][i] +(j-i)*S

&a[i][j]=&a[0][0] +[i*(2b+1)+(j-i)]*S

2 稀疏矩阵

稀疏矩阵：矩阵中大部分元素是零。

如何节省存储空间，不对零元素进行存储？

- 用三元组数组表示稀疏矩阵
- 用十字链表表示稀疏矩阵

jyang
@cs.ecnu.edu.cn

第四章 数组

稀疏矩阵的应用实例：本文检索。
数据：大量的document，若干个Query。
目标：对于每个query输出前k个最相关的document。

Document1： The weather is good today…
Document2： The movie is quite interesting. Jack like the movie very much…
Document3： Wind power is a clean energy source……Wind power is very useful in some district where wind is always strong….

	cloud	movie	sick	weather	name	year	wind	nation	power	week	…
Doc1：	0	0	0	1	0	0	0	0	0	0	…
Doc2：	0	2	0	0	0	0	0	0	0	0	…
Doc3:	0	0	0	0	0	0	3	0	2	0	…
……	…	…	…	…	…	…	…	…	…	…	…

jyang
@cs.ecnu.edu.cn

第四章 数组

稀疏矩阵的应用实例：本文检索

数据：大量的document，若干个Query。

目标：对于每个query输出前20个或前100个最相关的document。

Query1： Find movie reviews of the film "Les Miserables".

Query2： What are the pros and cons of using wind power.

	cloud	movie	sick	weather	name	year	wind	nation	power	week	…
Query1：	0	1	0	0	0	0	0	0	0	0	…
Query2：	0	0	0	0	0	0	1	0	1	0	…
……	…	…	…	…	…	…	…	…	…	…	…

jyang
@cs.ecnu.edu.cn

第四章 数组

稀疏矩阵的应用实例：本文检索
数据：大量的document，若干个Query。
目标：对于每个query输出前20个或前100个最相关的document。

表示：Document集和Query集分别用一个稀疏矩阵表示。
问题转换为：对query-document相关度进行打分和排序（多种算法）

最简单的方法：query-document中出现了同一单词，将得分+1。

Lexicon	cloud	movie	sick	weather	name	year	wind	nation	power	week	…
Query1 :	0	1	0	0	0	0	0	0	0	0	…
Query2 :	0	0	0	0	0	0	1	0	1	0	…
……	…	…	…	…	…	…	…	…	…	…	…

Lexicon	cloud	movie	sick	weather	name	year	wind	nation	power	week	…
Doc1:	0	0	0	1	0	0	0	0	0	0	…
Doc2:	0	2	0	0	0	0	0	0	0	0	…
Doc3:	0	0	0	0	0	0	3	0	2	0	…
……	…	…	…	…	…	…	…	…	…	…	…

2 稀疏矩阵

用三元组数组表示稀疏矩阵

用三元组(i, j, a_{ij})表示位于i行j列的非零元素。

按行号的递增次序（同一行按列号的递增次序）存放在一个由三元组组成的数组中。

2 稀疏矩阵

用三元组数组表示稀疏矩阵

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 12 & 15 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 36 & 46 & 0 & 52 \\ 0 & 0 & 0 & 0 \\ 0 & 72 & 0 & 68 \end{bmatrix} \end{array}$$

a	0	1	2
0	5	4	7
1	0	0	12
2	0	1	15
3	2	0	36
4	2	1	46
5	2	3	52
6	4	1	72
7	4	3	68

总行数　总列数　非零元素的个数

用三元组数组表示稀疏矩阵——矩阵转置

矩阵转置：对表示非零元素的三元组的行号和列号对调。

(i, j, a_{ij}) 变为 (j, i, a_{ij})
并把 (j, i, a_{ij})插在转置后结果矩阵的三元组数组的适当位置

要求：转置后的矩阵仍然按照行号的递增次序存储。

如何把新的元组插在适当的位置上?

jyang
@cs.ecnu.edu.cn

用三元组数组表示稀疏矩阵——矩阵转置

首先，在原矩阵的三元组数组a中寻找列号为0所有元素，并把它们作为行号为0的元素依次存放到表示转置矩阵的三元组数组b中；然后在a中寻找列号为1的所有元素……

```
q=1;
for  (col=0; col<n;col++) //n:总列数
     for (p=1;p<=t;p++)    //t:三元组的个数
       if (a[p][1]==col)    //依次寻找第0到第n-1列的元组，
       {                      //找到列号为col的，插入到b数组。
              b[q][0]=a[p][1];
              b[q][1]=a[p][0];
              b[q][2]=a[p][2];
              q++;
       }
```

用三元组数组表示稀疏矩阵——矩阵转置

```
q=1;
for  (col=0; col<n;col++)
     for (p=1;p<=t;p++)
      if (a[p][1]==col)
      {
            b[q][0]=a[p][1];
            b[q][1]=a[p][0];
            b[q][2]=a[p][2];
            q++;
      }
```

执行时间：主要花费在两重循环上

O(n*t)

当t=m*n，执行时间为$O(mn^2)$

2 稀疏矩阵

用三元组数组表示稀疏矩阵——矩阵转置

首先，确定a中每一列非零元素的个数，也就是b中每一行非零元素的个数。然后求出b中每一行第一个非零元素的存放位置。

a	0	1	2
0	5（行数）	4（列数）	7（个数）
1	0	0	12
2	0	1	15
3	2	0	36
4	2	1	46
5	2	3	52
6	4	1	72
7	4	3	68

b	0	1	2
0	4（行数）	5（列数）	7（个数）
1			
2			
3			
4			
5			
6			
7			

2 稀疏矩阵

用三元组数组表示稀疏矩阵——矩阵转置

a	0	1	2
0	5（行数）	4（列数）	7（个数）
1	0	0	12
2	0	1	15
3	2	0	36
4	2	1	46
5	2	3	52
6	4	1	72
7	4	3	68

a	0	1	2
0	4（行数）	5（列数）	7（个数）
1			
2			
3			
4			
5			
6			
7			

存放2个原矩阵中列号为0的元素

存放3个原矩阵中列号为1的元素

存放2个原矩阵中列号为3的元素

列号为0的元素：2个　x[0]=2
列号为1的元素：3个　x[1]=3
列号为2的元素：0个　x[2]=0
列号为3的元素：2个　x[3]=2

原矩阵中列号为0的元素存放的开始位置：1
……1……：1+x[0]：3
……2……：3+x[1]：6
……3……：6+x[2]：6

用三元组数组表示稀疏矩阵——矩阵转置

```
.....................
for (i=0;i<n;i++)   x[i]=0;
for (i=1;i<t;i++)   x[a[i][1]]+=1;
y[0]=1;
for (i=1;i<n;i++)   y[i]=y[i-1]+x[i-1];
for (i=1;i<t;i++)
{
      j=y[a[i][1]];
      b[j][0]=a[i][1];
      b[j][1]=a[i][0];
      b[j][2]=a[i][2];
      y[a[i][1]]=j+1;
}
```

第一个循环：执行n次
第二个循环：执行t次

第三个循环：执行（n-1）次
第四个循环：执行t次

当t=m*n，执行时间为O(m*n)

用十字链表表示稀疏矩阵

在矩阵运算过程中，如果非零元素的个数有变化，会引起数组中元素的移动。

十字链表结构：用一个结点表示稀疏矩阵的一个非零元素，在结点中，用row，col和val字段分别存放非零元素的行号、列号和元素的值；
Right指针：指向表示同一行中下一个非零元素的结点；
Cown指针：指向表示同一列中下一个非零元素的结点；

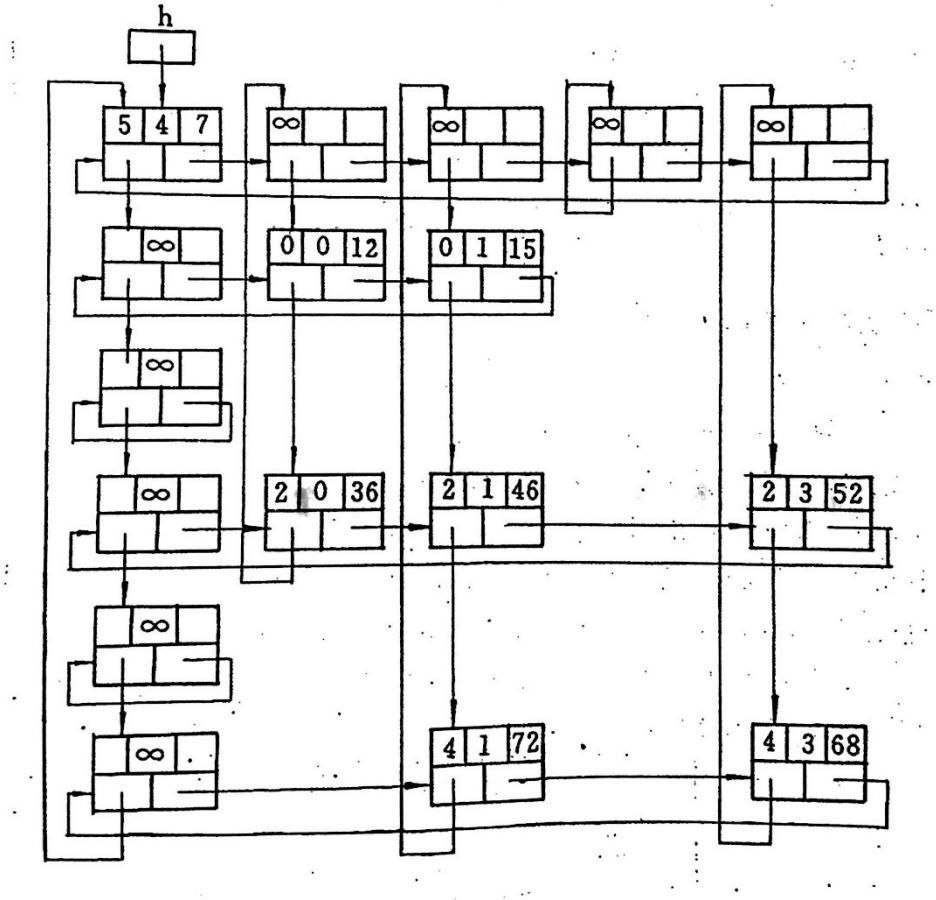
每一行，每一列，构成环形链表；一个结点即位于它所在行的环形链表中，又位于它所在列的环形链表中，行链表和列链表在位于该行和该列的结点处构成“十”字。

增加表头结点以便于处理。

2 稀疏矩阵

用十字链表表示稀疏矩阵

$$
\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{c}
\begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\
\begin{bmatrix} 12 & 15 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 36 & 46 & 0 & 52 \\ 0 & 0 & 0 & 0 \\ 0 & 72 & 0 & 68 \end{bmatrix}
\end{array}
$$



小结

数组及矩阵的存储