

# 稀疏矩阵的三元组表示法和十字链表法

## 稀疏矩阵

一个矩阵中有大量的零元素。对于稀疏矩阵，如果单纯的用一般矩阵来表示的话很浪费空间，因为非零元素仅有少量几个，我们可以用其他方式来表示少数非零元素在矩阵的位置就可以保存稀疏矩阵的全部信息了。

## 三元组

矩阵中每一个元素都是有行序号、列序号以及该元素的值唯一表示。因此，我们可以用三项内容唯一表示稀疏矩阵的每一个非零元素，即形式为：(i, j, value)，那么剩下位置都为零元素。

三元组表示法类型定义需要两个结构体来定义，一个用来保存每一非零元素的位置和值的信息，另一用来定义整个矩阵的信息，包括矩阵的行数和列数，非零元素的个数以及整个三元组表。

三元组结点定义：

```
#define MAX_SIZE 101
typedef int elemtype ;
typedef struct
{ int row ; /* 行下标 */
  int col ; /* 列下标 */
  elemtype value; /* 元素值 */
} Triple ;
```

三元组顺序表定义：

```
三元组顺序表定义
typedef struct
{ int rn ; /* 行数 */
  int cn ; /* 列数 */
  int tn ; /* 非0元素个数 */
  Triple data[MAX_SIZE] ;
} TMatrix ;
```

## 稀疏矩阵的转置——三元组表示

基本思想是

- 1、将矩阵的行、列下标值交换。即将三元组中的行、列位置i、j相互交换。
- 2、重排三元组表中的元素的顺序。即交换后仍然是按行优先顺序的。

时间复杂度为 $O(nmt)$ 其中t为非零个数。可见，但非零个数比较大时不适合用三元组表示法。

## 稀疏矩阵的快速转置——三元组表示

基本思想：直接按照稀疏矩阵**A**的三元组**a.data**的次序依次顺序转换，并将转换后的三元组放置于三元组表**b.data**的恰当位置。

比如对于行*i*的元素可以直接放在**b.data**的*i*的位置，但若*i*行有多个元素时，那么需要辅助的数组来记录每一行元素的个数，那么后续*i*行的元素可以放在 **b.data**的 *i*+1 的位置。

辅助向量有两个：

- num[col]：统计A中第col列中非0元素的个数
- cpot[col]：指示A中相应列第一个非0元素在b.data中的位置

显然有位置关系为：

$$\begin{cases} \text{cpot}[1]=1 \\ \text{cpot}[\text{col}]=\text{cpot}[\text{col}-1]+\text{num}[\text{col}-1] & 2 \leq \text{col} \leq \text{a.cn} \end{cases}$$

$$A=\begin{pmatrix} 0 & 12 & 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 24 & 0 & 0 & 2 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & -6 & 0 & 0 & 0 & 0 \end{pmatrix}$$

稀疏矩阵示例

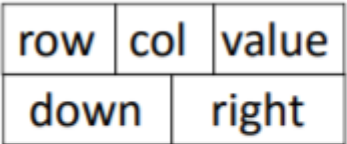
num[col]和cpot[col]的值表

col	1	2	3	4	5	6	7	8
num[col]	1	2	2	1	0	1	1	1
cpot[col]	1	2	4	6	6	7	8	9

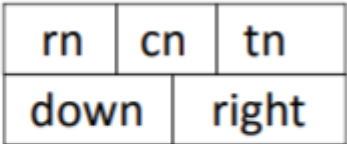
## 十字链表

对于稀疏矩阵，当非0元素的个数和位置在操作过程中变化较大时，采用链式存储结构表示比三元组的线性表更方便。

矩阵中非0元素的结点所含的域有：行、列、值、行指针 (指向同一行的下一个非0元)、列指针(指向同一列的下一个非0元)。其次，十字交叉链表还有一个头结点，结点的结构如图5-10所示。



(a) 结点结构



(b) 头结点结构

### 十字链表结点结构

由定义知，稀疏矩阵中同一行的非0元素的由right指针域链接成一个行链表， 由down指针域链接成一个列链表。则每个非0元素既是某个行链表中的一个结点，同时又是某个列链表中的一个结点，所有的非0元素构成一个十字交叉的链表，称为十字链表

此外，还可用两个一维数组分别存储行链表的头指针和列链表的头指针。

十字链表的结点描述如下：

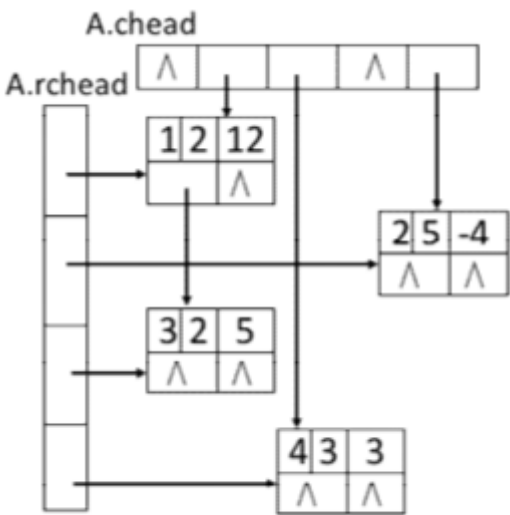
```
typedef struct Clnode
{
    int row , col ; /* 行号和列号 */
    elemtype value ; /* 元素值 */
    struct Clnode *down , *right ;
} OLNode ; /* 非0元素结点 */
```

```
typedef struct
{
    int rn; /* 矩阵的行数 */
    int cn; /* 矩阵的列数 */
    int tn; /* 非0元素总数 */
    OLNode *rhead ;
    OLNode *thead ;
} CrossList ;
```

举个例子，假如有如下稀疏矩阵：

$$A = \begin{pmatrix} 0 & 12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -4 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \end{pmatrix}$$

那么它的十字链表表示为：



(b) 稀疏矩阵的十字交叉链表

## 稀疏矩阵的转置——十字链表

十字链表下的稀疏矩阵转置十分简单，只要每个结点 row 和 col 交换以及down和right指针指向交换就完成转置。