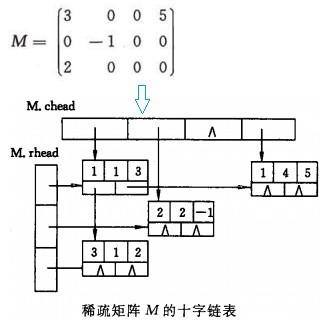
## [稀疏矩阵的十字链表存储](http://www.cnblogs.com/zhuyf87/archive/2012/10/22/2733628.html)

稀疏矩阵的压缩存储有几种方式，如：三元组顺序表、行逻辑链接的顺序表和十字链表。

使用链表存储的好处是：便于矩阵中元素的插入和删除。

例如：“将矩阵B加到矩阵A上”，那么矩阵A存储的元素就会有变动。比如会增加一些非零元，或者删除一些元素（因为bij+aij=0）。

下图是矩阵M和M的十字链表存储：



十字链表及其结点可用如下结构体表示：

|  |
| --- |
| typedef struct OLNode{  int i, j; // 非零元的行列下标 ElemType e;  struct OLNode \*right, \*down; // 向右域和向下域  } OLNode, \*OLink;  typedef struct{  OLink \*rhead, \*chead; // 行链表和列链表的头指针数组  int mu, nu, tu; // 稀疏矩阵的行数、列数和非零元个数  } CrossList; |

在通过代码创建十字链表时，要特别注意right、down和rhead、chead这些指针的赋值。

现在来看“将矩阵B加到矩阵A上”这个问题。所要做的操作：aij +bij，其结果一共会有4种情况：

1. aij（bij = 0）（不做变化）
2. bij（aij = 0）（在A中插入一个新结点）
3. aij +bij ≠ 0 （改变结点aij的值域）
4. aij +bij = 0 （删除结点aij）

假设指针pa和pb分别指向矩阵A和B中行值相同的两个结点，对于上述4种情况的处理过程为：

1. 若“pa == NULL”或“pa->j大于pb->j”，则在矩阵A中插入一个值为bij的结点。并且需要修改同一行前一结点的right指针，和同一列前一结点的down指针。
2. 若“pa->j小于pb-j”，则pa指针右移一步。
3. 若“pa->j等于pb-j”，并且“pa->e + pb->e != 0”，则修改pa->e即可。
4. 若“pa->j等于pb-j”，并且“pa->e + pb->e == 0”，则需要删除矩阵A中pa所指结点。并且需要修改同一行前一结点的right指针，和同一列前一结点的down指针。