数组

多维数组采用顺序存储时具有随机存储（计算各个元素存储位置的时间相等）特性

二维数组有两种存储方式：

- 以列序为主序
- 以行序为主序

数组元素存储位置的计算

根据下标求m行n列二维数组A中任意元素aij的存储位置；每个数据元素占d个存储单元

行列下标从1开始

- 以行序为主序；LOC(aij) = LOC(a11) + [(i-1)*n + (j-1)]*d
- 以列序为主序；LOC(aij) = LOC(a11) + [(j-1)*m + (i-1)]*d

行列下标从0开始

- 以行序为主序；LOC(aij) = LOC(a00) + (i*n + j)*d
- 以列序为主序；LOC(aij) = LOC(a00) + (j*m + i)*d

矩阵的压缩存储

有时为了节省存储空间，可以对有许多相同值或零值的矩阵进行压缩存储

特殊矩阵

值相同的元素或者零元素在矩阵中的分布有一定规律，我们称此类矩阵为特殊矩阵；

特殊矩阵-对称矩阵

若n阶矩阵A中的元满足下述性质 `aij = aji (1<=i,j<=n)` 则称为n阶对称矩阵

对于对称矩阵，我们可以为每一对对称元分配一个存储空间，则可将n^2个元压缩存储到 `[n(n+1)]/2` 个元的空间中

假设以一维数组sa[`[n(n+1)]/2`]作为n阶对称矩阵A的存储结构，则sa[k]和矩阵元aij之间存在着一一对应的关系

- 当i >= j；k = `[i(i-1)/2] + j - 1`
- 当i < j； k = `[j(j-1)/2] + i - 1`

特殊矩阵-三角矩阵

这种压缩存储的方法同样也适用于三角矩阵；

所谓上（下）三角矩阵是指矩阵的下（上）三角（不包括对角线）中的元均为常数c或0的n阶矩阵；

除了和对称矩阵一样只存储其上（下）三角中的元之外，再加一个存储常数c的存储空间即可

稀疏矩阵

值相同的元素或者零元素在矩阵中的分布没有规律，我们称此类矩阵为稀疏矩阵；

没有准确的定义，一般是指非零元素或值不相同元素较少的矩阵

按照压缩存储的概念，只需要存储稀疏矩阵的非零元；因此，除了存储非零元素的值之外，还必须同时记下它所在的行和列的位置（i，j）

反之，一个三元组（i，j，aij）唯一的确定了矩阵A的一个非零元；

因此，稀疏矩阵可由表示非零元的三元组以及行数唯一确定

```
typedef struct {
    int         i;          //行下标
    int         j;          //列下标
    ElementType element;    //元素值
} Triple;

typedef struct {
    Triple  data[MAX_SIZE];     //三元组
    int     row;                //行数
    int     column;             //列数
    int     notZeroNumber;      //非零元个数
} Matrix;
```

三元组的转置

```
//矩阵转置
Triple *transposeMatrix(Matrix &matrix){

    Triple *triples = (Triple *)malloc(sizeof(Triple) * matrix.notZeroNumber);

    //该变量用来记录新三元组数组的下标
    int index = 0;
    //columnCursor列游标
    int columnCursor;

    for(columnCursor = 0;columnCursor < matrix.column;columnCursor++){
        //序号游标 游标 可以移动的序号
        int indexCursor;

        for(indexCursor = 0;indexCursor < matrix.notZeroNumber;indexCursor++){
            //如果序号游标位置的三元组的列号与列游标相同
            //行列互换后 将其存入新的三元组数组
            //原本的三元组行号就是递增的 互换后成为列号还是递增的 所以不用对列再排序
            if(matrix.data[indexCursor].j == columnCursor){
                //行列互换
```

```
                Triple triple;
                triple.i = matrix.data[indexCursor].j;
                triple.j = matrix.data[indexCursor].i;
                triple.value = matrix.data[indexCursor].value;

                //存入新的三元组数组 index标志着它再新数组中的位置
                triples[index++] = triple;
            }
        }
    }

    return triples;
}
```

广义表

故名思意，广义表是线性表的推广；也有人称其为列表（lists，用复数形式以示与统称的list的区别）

广义表一般记作 `LS = (a1, a2, ..., an)`

其中LS是广义表(a1, a2, ..., an)的名称，n是它的长度

当广义表LS非空时

- 称第一个元素a1为LS的表头（Head）
- 称其余元素组成的表(a2, ..., an)时LS的表尾（Tail）