数组：

特点：结构中的元素本身可以是具有某种结构的数据，但是各元素应属于同一种数据类型。

数组具有以下性质：

1. 数据中的元素个数固定，一旦定义了一个数组，其元素个数不再有增减变化
2. 数组中每个数据元素都具有相同的数据类型
3. 数组中每个元素都有一组唯一的的下表与之对应，并且数组元素的下标具有上、下界约束且下标有序
4. 数组是一种随机存储结构，可随机存储数组中的任意元素

由于计算机的内存结构是一维的，因此多位数组的存储就必须按某种方式进行降维处理并最终由一维数组定义；因此，可通过递推关系将多维数组的数据元素转化为线性序列来存储。

特殊矩阵：

1. 对称矩阵：满足aij = aji
2. 三角矩阵：以主对角线来划分，分为上三角和下三角。上三角矩阵是指矩阵的下三角均为常数或0的n阶矩阵
3. 对角矩阵：除了主对角线上和主对角线两侧的若干对角线上的元素外，其他所有元素都为0。又称带状矩阵

稀疏矩阵：矩阵含有少量非零元素，但分布没有规律

1. 稀疏矩阵的三元组表示：

为了节省空间，其存储方式为压缩存储方式，即只存非零元素。由于元素无规律可循，所以每一个非零元素aij，要存储（i,j,aij）即（行，列，值）

还要用（m,n,t）记录稀疏矩阵的（行数，列数，非零元素总数）

1. 稀疏矩阵的十字链表表示：

基本思想：把每个非零元素存储为一个结点，由五部分组成

|  |  |  |  |
| --- | --- | --- | --- |
| row（所在行） | col（所在列） | | v（值） |
| down（指向同一列下一个非零元素） | | right（指向同一行下一个非零元素） | |

头结点：

|  |  |  |  |
| --- | --- | --- | --- |
| row（所在行） | col（所在列） | | \*next |
| down（指向同一列下一个非零元素） | | right（指向同一行下一个非零元素） | |

所以：

typedef struct node

{

int row,col;

struct node \*right,\*down;

union //共用体，说明v和\*next共用同一块内存，不能同时存在

{

int v;

struct node \*next;

}tag;

}MNode;

**广义表**

允许表中的元素，既可以是单元素，又可以是另外一个表。

与线性表的区别：

线性表的每个元素都是结构上不可再分的单元素

本身是带有参差的非线性结构，故用链式存储结构

孩子兄弟表示法：

设立一个flag，为1表示本结点有孩子结点，则没有数据域；

|  |  |  |
| --- | --- | --- |
| flag = 1 | childlist | next |

为0表示无孩子结点，有数据域

|  |  |  |
| --- | --- | --- |
| flag = 0 | data | next |

代码：

typedef struct node

{

int flag;

union

{

char data;

struct node \*childlist;

}val;

struct node \*next; //指向相邻后续结点的指针

}1snode,\*plsnode;

非线性结构，其长度是最外层包含的元素个数

深度：广义表中所包含括号的层数

通常用大写字母表示广义表，用小写字母表示单元素，用（），括起来，括号内的数据用逗号隔开

常用表示方法：

1. E=() 空表，长度为0
2. L=(a,b) 长度为2的广义表，每个元素都是原子，此时为线性表
3. A=(x,L)=(x,(a,b)) 长度为2的广义表，第一个元素为原子x，第二个为子表L
4. B=(A,y)=((x,(a,b)),y) 长度为2，第一个是子表，第二个是原子y
5. C=(A,B)=((x,(a,b)), ((x,(a,b)),y)) 长度为2，都是子表

带名字的广义表表示：

1. E()
2. L(a,b)
3. A(x , L(a,b))
4. B(A(x ,(a,b)) , y)
5. C(A (x, (a,b)) , B ((x, (a,b)), y))