第五章 数组与广义表

要求、目标：

了解数组的基本基本概念、性质；

掌握二维数组以行序和列序为主序时 a_{ij} 元素存储地址的计算；

掌握矩阵的压缩存储、稀疏矩阵的三元组存储

掌握广义表的长度深度计算，并能通过表头表尾函数求出对应元素

▶ 数 组

一、概述

1．数组：具有相同数据类型的若干元素构成的有序集合

2．性质：

①元素个数固定；

②元素类型相同；

③每个元素有唯一的下标值；

④是随机存取结构

3．二维数组的表示形式（m×n）

- 二维数组可看作是一个线性表，其每个元素也是一个线性表
- 多维数组的基本操作是存取元素和修改元素值，不能进行插入和删除操作

$$A_{m\times n}=\begin{bmatrix} a_{0,0} & \cdots & a_{0,n-1} \\ \vdots & & \vdots \\ a_{m-1,0} & \cdots & a_{m-1,n-1} \end{bmatrix}$$

二、数组的存储（m×n）

1．一维数组的存储

设有定义#define 100

int a[M]; 则 a[i]元素的地址为：

$Loc(a_i)=Loc(a_0)+i*d$ 其中 d 为每个元素所占的字节数，0≤i＜M

2．二维数组的存储

1）以行序为主序—a[i][j]的元素地址计算

①下标从 0 开始

$loc(a_{i,j})=loc(a_{0,0})+(i*n+j)*d$

例：设有 float a[5][4]; 起始地址为 2000，求 a[3][2]的地址

$loc(a_{3,2})=loc(a_{0,0})+(i*n+j)*d=2000+(3*4+2)*4=2056$

②下标不从 0 开始(A[c1…d1,c2…d2]) （ps：下标不从 0 难度稍大，但也要掌握）

$loc(a_{i,j})=loc(a_{c1,c2})+[(i-c1)*(d2-c2+1)+(j-c2)]*d$

例：设有 float a[-20…30,-30…20], 起始地址为 200，求 a[25][18]的地址

$loc(a_{25,18})=loc(a_{-20,-30})+[(25+20)*(20+30+1)+(18+30)]*4$

$=200+[45*51+48]*4$

$=9572$

2）以列序为主序—a[i][j]的元素地址计算

①下标从 0 开始

$loc(a_{i,j})=loc(a_{0,0})+(j*m+i)*d$

例：设有 float a[5][4]; 起始地址为 2000，求 a[3][2]的地址

$loc(a_{3,2})=loc(a_{0,0})+(j*n+i)*d=2000+(2*5+3)*4=2052$

②下标不从 0 开始(A[c1…d1,c2…d2])

$loc(a_{i,j})=loc(a_{c1,c2})+[(j-c2)*(d1-c1+1)+(i-c1)]*d$

例：设有 float a[-20…30,-30…20], 起始地址为 200，求 a[-18][-25]的地址

$loc(a_{-18,-25})=loc(a_{-20,-30})+[(-25+30)*(30+20+1)+(-18+20)]*4$

$=200+[5*51+2]*4$

$=1228$

★★★Key：无论矩阵元素下标是从 0 开始还是从几开始，关键在于按照列优先或者行优先的原则，去计算 aij 是第几个元素，再转化成数组元素，再计算地址。

三、特殊矩阵的压缩存储

1．压缩存储：为多个值相同的非零元素只分配一个存储空间，对零元素不分配空间

2．特殊矩阵：值相同的非零元素或零元素的分布有一定规律的矩阵。包括对称矩阵、三角矩阵、对角矩阵。

3．对称矩阵：若一个 n 阶方阵 A 中的元素满足 $a_{i,j}=a_{j,i}$ $(0\leqslant i,j\leqslant n-1)$则称其为 n 阶对称矩阵

4．对角矩阵：若一个 n 阶方阵满足其所有非零元素都集中在以主对角线为中心的带状区域中，则称其为 n 阶对角矩阵

5．三角矩阵的压缩存储（非对称）

1）下三角矩阵：上三角（不包括主对角线）中的元素都是 0 的 n 阶方阵

2）压缩存储方法

①只存储下三角的元素，其余的 0 不存储

②下三角矩阵的 n^2 个元素压缩到 $\frac{n(n+1)}{2}$ 个元素的存储单元中，可用一维数组 s[0…n(n+1)/2]存放下三角矩阵中的元素

元素个数：$1+2+3+\ldots+n=\frac{n(n+1)}{2}$

③按行序为主序进行存储

3）$a_{i,j}$ 的地址计算(行序 i≥j)

①a[i][j]与 s[k]的对应关系（下标均从 0 开始）

$$k=\begin{cases}\frac{i(i+1)}{2}+j & i\geq j\\ \frac{n(n+1)}{2} & i<j\end{cases}$$

当上三角为常数 C 时用此单元存放该常数

②计算地址

$loc(a_{i,j})=loc(a_{0,0})+[\frac{i(i+1)}{2}+j]*d$

例：下三角矩阵 A[1…8，1…8]按行存储，起始地址为 1000，每个元素占 4 个字节，求 A[7,5]的地址

$loc(a_{7,5})=loc(a_{0,0})+[\frac{(i-1)((i-1)+1)}{2}+(j-1)]*d$

=1000+[6*(6+1)/2+4]*4

=1100

6．对角矩阵的压缩存储

1）特点：非零元素分布在主对角线和其它对角线上，其余元素为 0

2）半带宽：主对角线上面或下面的对角线条数，记为 b

3）带宽：所有对角线的条数，记为 2b+1

4)m 对角矩阵含义：带宽为 m

5）当 0≤i,j≤n-1 且|i-j|>b 时，其元素 $a_{i,j}=0$

6）压缩存储

①只存储带状区域中的(2b+1)n-b(b+1)个元素，其余区域外的元素不存储

②按行存储，第 1 行、最后一行存 b+1 个元素，其余各行当作有(2b+1)个元素，若不中 2b+1 则在行前或行后补上一些元素，补上的位置空留，共需(2b+1)n-2b 个存储单元，多用 b(b-1)个单元

7）地址计算（行优先）

①用一维数组来存放带状矩阵，则 s[k]与 $a_{i,j}$ 对应关系：k=i(2b+1)+j-i（下标均从 0 开始）

②计算 $a_{i,j}$ 的地址

$loc(a_{i,j})=loc(a_{0,0})+[i(2b+1)+(j-i)]*d$

例：将 A[1…100，1…100]的三对角矩阵按行优先存入一维数组 B[1…298]中，A 中元素 $a_{66,65}$ 在 B 中的位置 k 为多少？

k=i(2b+1)+j-i=(66-1)(2*1+1)+((65-1)-(66-1))=194 即 k=194+1=195

四、稀疏矩阵

1．定义：大多数元素值为零，且非零元素的分布没有规律的矩阵

2．形式：（6×7）

$$\begin{bmatrix} 0 & 12 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 24 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & -7 & 0 & 0 & 0 \end{bmatrix}$$

3．稀疏矩阵的三元组表示

1）压缩方法

只存储非零元素，同时保存非 0 元素的行号、列号，用一个三元组（i,j,a_{ij}）表示一个非零元素，所有非零元素构成三元组线性表，按行号的递增次序存放在一个由三元组组成的二维数组中。既可以采用数组存储，也可以采用十字链表存储。

例：上面矩阵的三元组线性表为：

((0,1,12),(0,2,9),(2,0,-3),(2,5,14),(3,2,24),(4,1,18),(5,0,15),(5,3,-7))

2）三元组顺序表（不具有随机存取功能）：用顺序存储结构表示三元组表

3）三元组表示方法

若有 t 个非 0 元素，可定义一个(t+1)×3 的二维数组 a[t+1][3]，其中第 0 行分别存放矩阵的行数、列数和非 0 元素的个数，从第 1 到 t 行依次存放 t 个非 0 元素对应的 t 个三元组

例：上例中 t=8,可定义 a[9][3]，其存储形式为：

a	**0**	**1**	**2**
0	6	7	8
1	0	1	12
2	0	2	9
3	2	0	-3
4	2	5	14
5	3	2	24
6	4	1	18
7	5	0	15
8	5	3	-7

▶ 广义表

（1）广义表的概念

广义表是 n（n≥0）个数据元素 a1，a2，a3，…，an 的有序序列，一般记作：Ls=（a1，a2，...，an）。

其中：Ls 是广义表的名称，n 是它的长度，每个 ai（1≤i≤n）是 Ls 的成员，它可以是不可再分的单个元素（原子），也可以是一个可再分的广义表，分别称为广义表 Ls 的单元素和子表。

●表头：对于非空的广义表，第一个元素就是表头元素。就像上边的 a1，就是广义表的表头元素。 （注意不打括号！）

●表尾：对于非空的广义表，表尾元素就是除表头元素之外的其余所有元素组成的广义表。如（a2,...，ai，...，an)为 Ls 的表尾（tail）。 （注意要打括号！）

●表的长度：广义表的元素个数

●表的深度：表中元素的最深嵌套层数。

（2）广义表的基本运算（★重点★）

广义表有两个重要的基本操作，即取头操作（Head）和取尾操作（Tail）。★★★Key：一般 828 考试广义表部分主要针对用取头取尾操作如何从广义表中取出目标元素，同时对表的深度，长度也有所考察。

★★★怎么确定广义表的深度呢？可通过画树的简便方法来确定。