

## 数组和广义表

数组和广义表都是线性表结构的推广,在软件设计中得到广泛应用。本章主要学习数组的逻辑结构、存储结构以及矩阵的压缩存储等,还要学习广义表的逻辑结构、存储结构以及广义表基本操作算法。

### 5.1 知识点串讲

#### 5.1.1 知识结构图

本章重要的知识点结构如图 5.1 所示。

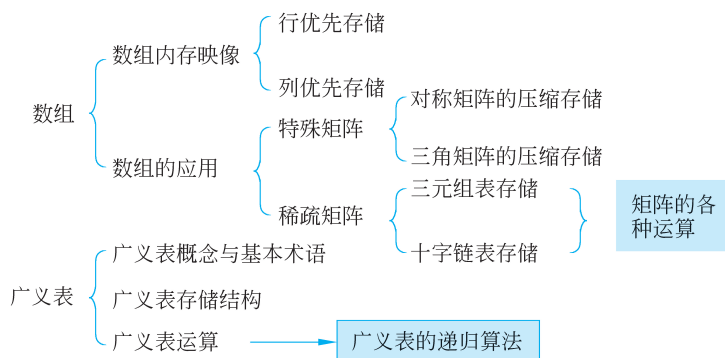


图 5.1 数组和广义表的知识结构图

#### 5.1.2 相关术语

- (1) 数组、二维数组、多维数组。
- (2) 特殊矩阵、稀疏矩阵。
- (3) 对称矩阵、三角矩阵、上三角矩阵、下三角矩阵。
- (4) 三元组表、十字链表。
- (5) 广义表、表头、表尾。
- (6) 递归表、广义表长度、广义表深度。

### 5.1.3 数组的存储结构

数组在内存被映像为向量,即用向量作为数组的一种存储结构。这是因为内存的地址空间是一维的,数组的行列固定后,通过一个映像函数,则可根据数组元素的下标得到它的存储地址。

#### 1. 以行为主序的顺序存储

以行为主序(或先行后列)的顺序存放,如 BASIC、PASCAL、COBOL、C 等程序设计语言中用的是以行为主的顺序分配,即一行分配完了接着分配下一行。

#### 2. 以列为主序的顺序存储

以列为主序(先列后行)的顺序存放,如 FORTRAN 语言中,用的是以列为主序的分配顺序,即一列一列地分配。

### 5.1.4 特殊矩阵

对三角矩阵、对称矩阵来说,可以用一维数组表示它们的压缩存储,以达到节省存储空间的目的。因此,实现特殊矩阵压缩存储的关键是找出它们之间的变换对应关系。

#### 1. 对称矩阵

对称矩阵的特点是:在一个  $n$  阶方阵中,有  $a_{ij} = a_{ji}$ ,其中  $0 \leq i, j \leq n-1$ 。对称矩阵关于主对角线对称,因此只需存储上三角或下三角部分即可。

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  $A$  中任一元素  $a_{ij}$  和  $sa[k]$  之间存在如下对应关系:

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

$a_{00}$  存储在  $sa[0]$ ,  $a_{10}$  存储在  $sa[1]$ ,  $\dots$ ,  $a_{(n-1)(n-1)}$  存储在  $sa[n(n+1)/2-1]$ 。

#### 2. 三角矩阵

三角矩阵中重复元素  $c$  可共享一个存储空间,其余的元素正好有  $n(n+1)/2$  个,可用一维数组  $sa[n(n+1)/2+1]$  作为  $n$  阶下(上)三角矩阵  $A$  的存储结构。

其中,常量  $c$  存放在数组的最后一个单元中,则当  $A$  为下三角矩阵时,任一元素  $a_{ij}$  和  $sa[k]$  之间存在如下对应关系:

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

### 5.1.5 稀疏矩阵

稀疏矩阵是零元素很多并且分布没有规律的矩阵,其压缩存储可采用三元组或者基于链式存储结构的十字链表方式。

#### 1. 三元组

将三元组按行优先的顺序,同一行中列号从小到大的规律排列成一个线性表,称为三元组表。一般采用顺序存储方法存储该表用来表示一个稀疏矩阵,存储三元组表的同时存储该矩阵的行、列,以及矩阵的非零元素的个数。

```
#define MAXSIZE 1000
typedef struct {
    int i, j;           /* 非零元素的行、列号 */
    DataType v;       /* 非零元素的值 */
}triple;              /* 三元组类型 */
typedef struct {
    triple data[MAXSIZE]; /* 非零元素的三元组表 */
    int m, n, t;         /* 稀疏矩阵的行数、列数和非零元素的个数 */
}tripletable;         /* 三元组表的存储类型 */
```

重点内容: 稀疏矩阵在压缩存储方式下的转置和相加等操作实现的算法。

#### 2. 十字链表

十字链表表示稀疏矩阵的方法是: 每个非零元素存储表示为一个结点,结点由 5 个域组成,其结构如图 5.2(a)所示。其中, row 域存储非零元素的行号, col 域存储非零元素的列号, value 域存储本元素的值, 指向本行中下一个非零元素行指针域 right 和指向本列下一个非零元素列指针域 down。同一行的非零元素通过 right 域链接成一个线性链表,同一列的非零元素通过 down 域链接成一个线性表,每个非零元素既是某个行链表中的一个结点,又是某个列链表中的一个结点,整个矩阵构成了一个十字交叉的链表,故称这样的链表为十字链表。



图 5.2 十字链表的结点结构

为便于操作,在十字链表的行链表和列链表上设置行头结点、列头结点和十字链表头结点。它们采用和非零元素结点类似的结点结构,具体如图 5.2(b)所示。其中行头结点和列头结点的 row 和 col 域值均为零;行头结点的 right 指针指向该行链表的第一个结点,它的 down 指针为空;列头结点的 down 指针指向该列链表的第一个结点,它的 right 指针为空。

结点的存储结构定义如下:

```
typedef struct node {
    int row, col;           /* 非零元素的行号和列号 */
    struct node * down, * right;
    union {
        DataType value;    /* 非零元素的值 */
        struct node * next;
    }
}crosslist;
```

### 5.1.6 广义表

#### 1. 广义表的基本概念

广义表是  $n \geq 0$  个元素  $a_1, a_2, \dots, a_n$  的有限序列, 即:

$$LS = (a_1, a_2, \dots, a_n)$$

其中, LS 是广义表的名称,  $n$  是它的长度。  $a_i$  可以是单个元素, 也可以是广义表。若  $a_i$  是单个元素, 则称它是广义表 LS 的原子; 若  $a_i$  是广义表, 则称它为 LS 的子表。当 LS 非空时, 称第一个元素  $a_1$  为 LS 的表头 (Head), 其余元素组成的表  $(a_2, a_3, \dots, a_n)$  为表尾 (Tail)。

广义表的深度是指广义表中所含括号的重数。

#### 2. 广义表的表示

由于广义表中的数据元素既可能是列表也可能是单元素, 相应的结点结构形式有两种: 一种是表结点, 用以表示列表; 另一种是原子结点, 用以表示单个原子。结点形式如图 5.3 所示。其数据结构定义如下:

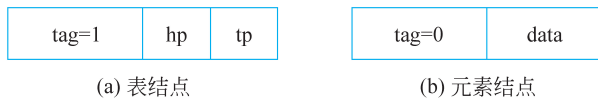


图 5.3 广义表结点示意图

```
typedef struct GenealNode {
    int tag;           /* 取值 0 或 1, 标志域, 用于区分原子结点和表结点 */
    union {           /* 原子结点和表结点的联合部分 */
        DataType data; /* 原子结点的值域 */
        struct {
            struct GenealNode * hp;
            struct GenealNode * tp;
        }ptr;
    }; /* ptr 是表结点的指针域, ptr.hp 和 ptr.tp 分别指向表头和表尾 */
} * GList;           /* 广义表类型 */
```

### 3. 广义表的操作

广义表上的操作除了拥有查找、取元素、插入和删除等基本操作外,还有几个特殊的基本操作:取表头、取表尾、求广义表的深度等。

## 5.2 典型例题详解

### 一、选择题

1. 下面的说法不正确的是( )。

- A. 数组是一种线性结构
- B. 数组是一种定长的线性表结构
- C. 除了插入与删除操作外,数组的基本操作还有存取、修改、检索、排序等
- D. 数组的基本操作有存取、修改、检索和排序等,没有插入和删除操作

**分析:** 数组的主要操作是存取、修改、检索和排序。数组没有插入和删除操作。答案应选择 C。

2. 一维数组 A 采用顺序存储结构,每个元素占用 6 字节,第 6 个元素的起始地址为 100,则该数组的首地址是( )。

- A. 64
- B. 28
- C. 70
- D. 90

**分析:** 设数组元素的首地址为  $x$ ,则存在关系  $x + (6-1) \times 6 = 100$ ,因此  $x$  为 70,答案应选择 C。

3. 稀疏矩阵采用压缩存储的目的主要是( )。

- A. 表达变得简单
- B. 对矩阵元素的存取变得简单
- C. 去掉矩阵中的多余元素
- D. 减少不必要的存储空间的开销

**分析:** 答案应选择 D。

4. 若对  $n$  阶对称矩阵 A 以行序为主序方式将其下三角形的元素(包括主对角线上所有元素)依次存放于一维数组  $B[1..(n(n+1))/2]$  中,则在 B 中确定  $a_{ij}$  ( $i < j$ ) 的位置  $k$  的关系为( )。

- A.  $i \times (i+1)/2 + j$
- B.  $j \times (j+1)/2 + i$
- C.  $i \times (i+1)/2 + j + 1$
- D.  $j \times (j+1)/2 + i + 1$

**分析:** 设以行为主序放对称矩阵的下三角元素,其存储结构如图 5.4 所示, $a_{00}$  存储在  $B[1]$ , $a_{10}$  存储在  $B[2]$ , $\dots$ , $a_{(n-1)(n-1)}$  存储在  $B[(n(n+1))/2]$ ,则位置  $k$  与  $(i, j)$  的对应关系为

$k = \begin{cases} i(i+1)/2 + j + 1, & i \geq j \\ j(j+1)/2 + i + 1, & i < j \end{cases}$ ,故答案应选择 C。

B[]	1	2	3	4	5	6	...	$k$	...	$(n(n+1))/2-1$	$(n(n+1))/2$
A	$a_{00}$	$a_{10}$	$a_{11}$	$a_{20}$	$a_{21}$	$a_{22}$	...	$a_{ij}$	...	$a_{(n-1)(n-2)}$	$a_{(n-1)(n-1)}$

图 5.4 对称矩阵的存储示意图

5. 已知广义表  $LS = ((a, b, c), (d, e, f))$ , 运用  $GetHead$  和  $GetTail$  运算取出  $LS$  中的元素  $e$  的运算是( )。

- A.  $GetHead(GetTail(LS))$
- B.  $GetHead(GetTail(GetTail(GetHead(LS))))$
- C.  $GetTail(GetHead(LS))$
- D.  $GetHead(GetTail(GetHead(GetTail(LS))))$

**分析:** 本题的解答过程要应用排除法, 分别对 A、B、C、D 选项进行计算并判断。根据选项 D 可知:  $GetHead(GetTail(GetHead(GetTail(LS)))) = GetHead(GetTail(GetHead((d, e, f)))) = GetHead(GetTail(d, e, f)) = GetHead((e, f)) = e$ 。答案应选择 D。

6. 设广义表  $L = ((a, b, c))$ , 则  $L$  的长度和深度分别为( )。

- A. 1 和 1
- B. 1 和 3
- C. 1 和 2
- D. 2 和 3

**分析:** 该题目主要考查广义表的长度和深度的基本概念, 广义表的长度是广义表中层次为 1 的元素个数, 而广义表的深度是指广义表展开后所含括号的层数。因此本题中的  $L$  的长度为 1,  $L$  的深度为 2。答案应选择 C。

7. 一个  $100 \times 90$  的稀疏矩阵, 非零元素有 10 个整型数, 设每个整型数占两字节, 则用三元组表示该矩阵时, 所需的字节数是( )。

- A. 60
- B. 66
- C. 18 000
- D. 33

**分析:** 三元组表结构为(行、列、元素值), 用三元组表表示稀疏矩阵时, 还需记录稀疏矩阵的行数、列数以及非零元素个数, 本题中所需的字节数应为  $10 \times (1+1+1) \times 2 + 3 \times 2 = 66$ 。答案应选择 B。

## 二、判断题

1. 稀疏矩阵压缩后, 必会失去随机存取功能。

**答案:** 正确。

**分析:** 具有存取任一个元素的时间相等这一特点的存储结构称为随机存取结构。对稀疏矩阵压缩存储所用的存储结构是三元组表或十字链表。十字链表因其链表结构而不能随机存取; 而使用三元组表存储矩阵时, 若要访问元素  $a_{ij}$ , 则必须扫描三元组表, 显然, 查找三元组表中前后元素所消耗的时间不同。

2. 线性表可以看成广义表的特例, 如果广义表中的每个元素都是原子, 则广义表便成为线性表。

**答案:** 正确。

3. 广义表中的元素可以是一个不可分割的原子, 或者是一个非空的广义表。

**答案:** 错误。

**分析:** 该题目主要考查广义表的定义, 广义表中元素可以是原子, 也可以是表(包括空表和非空表)。

4. 广义表中原子个数即为广义表的长度。

**答案:** 错误。

**分析:** 该题目主要考查广义表长度的定义, 广义表的长度是广义表中层次为 1 的元素个数。因此本题说法是片面的。

5. 数组可看成线性结构的一种推广, 因此与线性表一样, 可以对它进行插入、删除等操作。

**答案:** 错误。

**分析:** 数组可以看成一种特殊的线性表, 数组在维数和上下界确定后, 其元素个数已经确定, 不能进行插入和删除运算。但线性表可以插入、删除。

### 三、填空题

1. 需要压缩存储的矩阵可分为\_\_\_\_\_矩阵和\_\_\_\_\_矩阵两种。

**答案:** 特殊、稀疏

**分析:** 关于矩阵的压缩存储方法, 根据矩阵的特点分为特殊矩阵和稀疏矩阵。特殊矩阵要求进行压缩存储时保持其随机存取特性不变, 而稀疏矩阵则会失去随机存取功能。

2. 设数组  $A[0..8, 1..10]$ , 数组中任一元素  $A[i][j]$  均占内存 48b, 从首地址 2000 开始连续存放在主内存里, 主内存字长为 16b, 那么:

(1) 存放该数组至少需要的单元数是 (1) 。

(2) 存放数组的第 8 列的所有元素至少需要的单元数是 (2) 。

(3) 数组按列优先存储时, 元素  $A[5][8]$  的起始地址是 (3) 。

**答案:** (1)270, (2)27, (3)2204

**分析:** (1) 数组  $A[0..8, 1..10]$  的数组元素个数为  $9 \times 10 = 90$ , 因此存放该数组的单元个数为  $90 \times 48 / 16 = 270$ , 其中每个单元占据  $48 / 16 = 3$  个字长。(2) 存放数组的第 8 列的所有元素应为  $9 \times 48 / 16 = 27$ 。(3) 若数组按列优先存储, 元素  $A[5][8]$  的起始地址为  $2000 + (7 \times 9 + 5) \times 3 = 2204$ 。

3. 设数组  $A[1..50, 1..80]$  的基地址为 2000, 每个元素占两个存储单元, 若以行序为主序顺序存储, 则元素  $A[45, 68]$  的存储地址为 (1); 若以列序为主序顺序存储, 则元素  $A[45][68]$  的存储地址为 (2)。

**答案:** (1)9174, (2)8878

**分析:** 本题主要考查计算某元素的存储地址, 注意区分按行为主序和以列为主序的存储方式。

(1)  $A[45][68]$  的存储地址 =  $2000 + ((45 - 1) \times 80 + 67) \times 2 = 9174$ 。

(2)  $A[45][68]$  的存储地址 =  $2000 + ((68 - 1) \times 50 + 44) \times 2 = 8788$ 。

4. 设广义表  $L = (( ), ( ))$ , 则  $\text{GetHead}(L)$  是\_\_\_\_\_ ;  $\text{GetTail}(L)$  是\_\_\_\_\_ ;  $L$  的长度是\_\_\_\_\_ ;  $L$  的深度是\_\_\_\_\_。

**答案:**  $( ), (( ))$ 、2、2

**分析:** 对于广义表  $LS = (a_1, a_2, a_3, \dots, a_n)$ , 各个运算如下:

表头:  $\text{GetHead}(LS) = a_1$ ;

表尾:  $\text{GetTail}(LS) = (a_2, a_3, \dots, a_n)$ ;





**分析与解答：**

(1) 稀疏矩阵的三元组表为

$s = ((1, 2, -3), (1, 3, 1), (2, 3, 1), (2, 6, -1), (4, 3, 2), (5, 5, 4))$ , 其中三元组分别表示非零元素行号、列号以及元素值。其顺序存储结构如图 5.5 所示。

	1	2	3
0	1	2	-3
1	1	3	1
2	2	3	1
3	2	6	-1
4	4	3	2
5	5	5	4
6	稀疏矩阵行数		
6	稀疏矩阵列数		
6	稀疏矩阵非零元素个数		

图 5.5 稀疏矩阵 A 的三元组表

(2) 十字链表法实际上是采用链式存储结构表示三元组表。在链表中, 每个非零元素可用一个含有 5 个域的结点表示。其存储结构定义如图 5.2 所示。

稀疏矩阵 A 所对应的十字链表存储结构如图 5.6 所示。

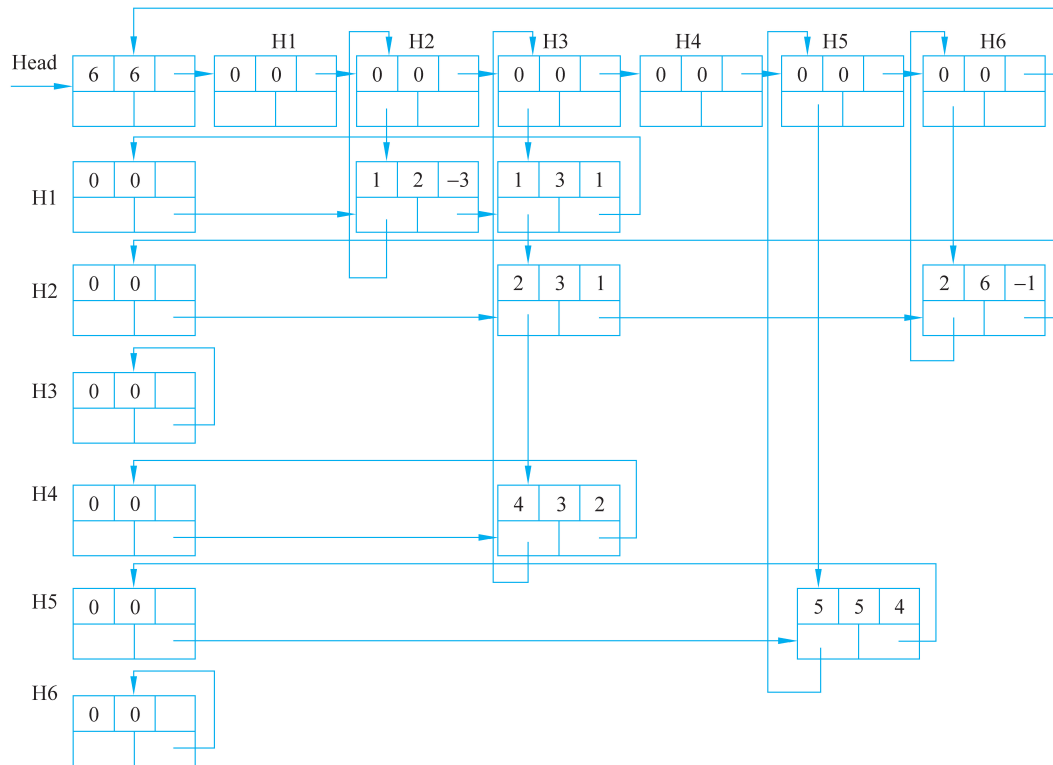


图 5.6 稀疏矩阵 A 的十字链表

4. 画出下列广义表的图形表示。

- (1)  $A(a, B(b, d), C(e, B(b, d), L(f, g, h)))$ ;
- (2)  $A(a, B(b, A))$ 。

**分析与解答：**

(1)  $A(a, B(b, d), C(e, B(b, d), L(f, g, h)))$  的图形表示如图 5.7(a) 所示。