华中科技大学

数据结构

第5章　数组和广义表

主讲教师：祝建华

引言：

线性表：$L=(a_1, a_2, ..., a_n)$，a_i是同类型的元素，$1 \leq i \leq n$

数组：　$A=（a_1, a_2, ..., a_n)$

若a_i是同类型的元素，A是一维数组，$1 \leq i \leq n$

若a_i是同类型的定长线性表，A是多维数组，$1 \leq i \leq n$

广义表：$Ls=（a_1, a_2, ..., a_n)$

a_i可以是同类型的元素或广义表，$1 \leq i \leq n$

5.1 数组的基本概念及其操作

数组是相同类型的数据的有限的、有序的组合。

5.1.1 数组的递归定义

1. 一维数组：

是一个定长线性表$(a_1, a_2, ..., a_n)$。

其中：a_i为数据元素，i为下标/序号，$1 \leqslant i \leqslant n$

$(a_1, a_2, ..., a_n)$又称为向量。

2.二维数组是一个定长线性表$(\alpha_1,\alpha_2,...,\alpha_m)$，
其中：$\alpha_i=(a_{i1},a_{i2},...,a_{in})$为行向量，$1\leqslant i\leqslant m$
由m个行向量组成，记作：

$$A_{m*n}=\begin{pmatrix} (\ a_{11} & a_{12} & \ldots & a_{1n}\ ) \\ (\ a_{21} & a_{22} & \ldots & a_{2n}\ ) \\ & \ldots\ldots\ldots\ldots & & \\ (\ a_{m1} & a_{m2} & \ldots & a_{mn}\ ) \end{pmatrix}$$

即 $A_{m*n}=((a_{11}\ a_{12}\ ...\ a_{1n}),(a_{21}\ a_{22}\ ...\ a_{2n}),...,(a_{m1}\ a_{m2}\ ...\ a_{mn}))$

或由n个列向量组成，记作：

$$A_{m\times n}=\begin{pmatrix} a_{11} & a_{12} & & a_{1n} \\ a_{21} & a_{22} & \vdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & & a_{mn} \end{pmatrix}$$

3. 三维数组是一个定长线性表($\beta_1, \beta_2, ..., \beta_p$)。

其中：β_k=($\alpha_1, \alpha_2, ..., \alpha_m$)为定长二维数组，$1 \leqslant k \leqslant p$

例　三维数组A[1..3, 1..4, 1..2]， p=3， m=4， n=2

$$A_{3*4*2} = \begin{pmatrix} a_{111} & a_{112} \\ a_{121} & a_{122} \\ a_{131} & a_{132} \\ a_{141} & a_{142} \end{pmatrix} \begin{pmatrix} a_{211} & a_{212} \\ a_{221} & a_{222} \\ a_{231} & a_{232} \\ a_{241} & a_{242} \end{pmatrix} \begin{pmatrix} a_{311} & a_{312} \\ a_{321} & a_{322} \\ a_{331} & a_{332} \\ a_{341} & a_{342} \end{pmatrix}$$

第1页　第2页　第3页

5.1.2 数组的操作

1. 生成一个数组： int a[7]；//生成静态一维数组
2. 赋值/修改
   a[1]=15； (a[1])++；
3. 取元素的值：
   a[0]=a[1]*2；
4. 销毁一个数组

a	32	16					
	0	1	2	3	4	5	6

5.1.3 程序设计举例

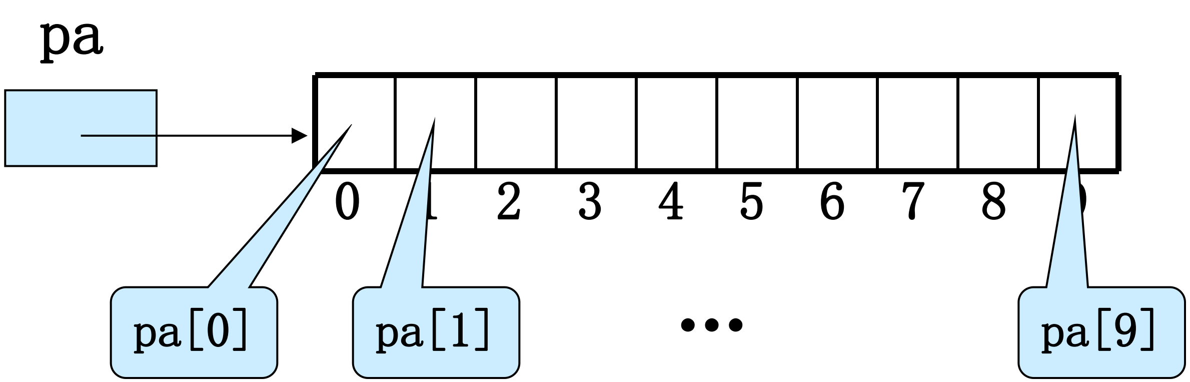
例1 main()

```
{ int i,a[10];                          //生成一维数组a
  for (i=0; i<10; i++)
     scanf(" %d",&a[i]);                //输入元素
  for (i=0; i<10; i++)
     printf(" %d ",a[i]*a[i]);          //输出元素的平方
}
```

退出时释放a

例2 生成动态的10个整数的一维数组

```
int   *pa;                                  //指针变量pa
pa=(int *)malloc(10*sizeof(int)); //动态数组pa
```



```
main()
{ int  i,n,*pa;
  scanf(” %d” ,&n);                    //动态输入n
  pa=(int *)malloc(n*sizeof(int)); //生成动态数组*pa
  for (i=0; i<n; i++)
    *(pa+i)=2*i;                  //指针法引用数组元素,赋值
  for (i=0; i<n; i++)
    printf( “%d, ” ,*(pa+i)); //输出数组元素0,2,4,6,...
  for (i=0; i<n; i++)
    scanf( “%d” ,&pa[i]);     //下标法引用数组元素，输入
  for (i=0; i<n; i++)
    printf(″%d,″,pa[i]);      //输出数组元素
  free(pa);                    //释放(销毁)数组空间
}
```

5.2数组的顺序表示和实现

5.2.1顺序表示(顺序存储结构)

1. 以行序为主序的顺序存储方式

左边的下标后变化，右边的下标先变化

2. 以列序为主序的顺序存储方式

左边的下标先变化，右边的下标后变化

例1 二维数组a[1..3,1..2],b是首地址,s是元素所占的单元数

$$\begin{pmatrix} a11 & a12 \\ a21 & a22 \\ a31 & a32 \end{pmatrix}$$

逻辑结构

以行序为主序

序号	内存	地址
1	a11	b
2	a12	b+s
3	a21	b+2*s
4	a22	b+3*s
5	a31	b+4*s
6	a32	b+5*s

以列序为主序

序号	内存	地址
1	a11	b
2	a21	b+s
3	a31	b+2*s
4	a12	b+3*s
5	a22	b+4*s
6	a32	b+5*s

例2 三维数组a[1..2, 1..3, 1..2]

第1页

$$\begin{pmatrix} a111 & a112 \\ a121 & a122 \\ a131 & a132 \end{pmatrix}$$

第2页

$$\begin{pmatrix} a211 & a212 \\ a221 & a222 \\ a231 & a232 \end{pmatrix}$$

逻辑结构

以行序为主序

序号	内存	地址
1	a111	b
2	a112	b+s
3	a121	b+2*s
4	a122	b+3*s
5	a131	b+4*s
6	a132	b+5*s
7	a211	b+6*s
8	a212	
9	a221	
10	a222	
11	a231	
12	a232	b+11*s

以列序为主序

序号	内存	地址
1	a111	b
2	a211	b+s
3	a121	b+2*s
4	a221	b+3*s
5	a131	b+4*s
6	a231	b+5*s
7	a112	b+6*s
8	a212	
9	a122	
10	a222	
11	a132	
12	a232	b+11*s

5.2.2数组的映象函数

数组元素的存储地址

例1 一维数组a[0..n-1]

a0	a1	a2	...	ai	...	a(n-1)	

下标	0	1	2	i	n-1
地址	b	b+s	b+2*s	b+i*s	b+(n-1)*s

设:b为首地址,s为每个元素所占的存储单元数

则:元素a[i]的存储地址：

$$Loc(i)=Loc(0)+i*s=b+i*s \qquad 0\leqslant i\leqslant n-1$$

例2 一维数组a[1..n]

	a1	a2	a3	...	ai	...	an	
下标	1	2	3		i		n	
地址	b	b+s	b+2s		b+(i-1)s		b+(n-1)s	

元素a[i]的存储地址

$$Loc(i)=Loc(1)+(i-1)*s=b+(i-1)*s \quad 1\leqslant i\leqslant n$$

例3 二维数组a[1..m, 1..n],假定无零行零列

$$A_{mxn}=\left(\begin{array}{ccccc} a_{11} & \ldots & a_{1j} & \ldots & a_{1n} \\ \multicolumn{5}{c}{\ldots\ldots\ldots\ldots\ldots\ldots} \\ a_{i1} & \ldots & a_{ij} & \ldots & a_{in} \\ \multicolumn{5}{c}{\ldots\ldots\ldots\ldots\ldots\ldots} \\ \underbrace{a_{m1} \quad \ldots}_{\text{共}j-1\text{列}} & & a_{mj} & \ldots & a_{mn} \end{array}\right)$$

共i-1行

(1) 以行序为主序，a[i][j]的地址为

$$Loc(i,j)=Loc(1,1)+(n*(i-1)+j-1)*s$$
$$=b+(n*(i-1)+j-1)*s \qquad 1\leqslant i\leqslant m, 1\leqslant j\leqslant n$$

其中： b为首地址，s为每个元素所占的存储单元数

n:列数　　m:行数

例3. 二维数组a[1..m,1..n],假定无零行零列

$$
A_{mxn}=\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
a_{m1} & a_{m2} & \cdots & a_{mj} & \cdots & a_{mn}
\end{pmatrix}
$$

共i-1行

共j-1列

(2) 以列序为主序，a[i][j]的地址为

$$Loc(i,j)=Loc(1,1)+(m*(j-1)+i-1)*s$$
$$=b+(m*(j-1)+i-1)*s \qquad 1\leqslant i\leqslant m,\ 1\leqslant j\leqslant n$$

其中： b为首地址，s为每个元素所占的存储单元数

n：列数　　m：行数

例4 二维数组a[0..m-1,0..n-1]（有零行零列）

$$A_{mxn}=\begin{pmatrix} a_{00} & \cdots & a_{0j} & \cdots & a_{0n-1} \\ a_{10} & \cdots & a_{1j} & \cdots & a_{1n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i0} & \cdots & a_{ij} & \cdots & a_{in-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \underbrace{a_{m-10} \ \cdots}_{\text{共j列}} & & a_{m-1j} & \cdots & a_{m-1n-1} \end{pmatrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} \text{共i行}$$

(1) 以行序为主序，a[i][j]的地址为

$$Loc(i,j)=Loc(0,0)+(n*i+j)*s$$
$$=b+(n*i+j)*s \qquad 0\leqslant i\leqslant m-1, 0\leqslant j\leqslant n-1$$

其中：b为首地址，s为每个元素所占的存储单元数

n：列数　　m：行数

例4 二维数组a[0..m-1, 0..n-1]（有零行零列）

$$A_{m\times n}=\begin{pmatrix} a_{00} & a_{01} \ldots & a_{0j} & \ldots & a_{0n-1} \\ a_{10} & a_{11} \ldots & a_{1j} & \ldots & a_{1n-1} \\ \ldots\ldots\ldots\ldots & & \ldots & & \ldots\ldots\ldots\ldots \\ a_{i0} & a_{i1} \ldots & a_{ij} & \ldots & a_{in-1} \\ \ldots\ldots\ldots\ldots & & \ldots & \ldots\ldots\ldots\ldots & \\ a_{m-10} & a_{m-11} \ldots & a_{m-1j} & \ldots & a_{m-1n-1} \end{pmatrix}$$

共i行

共j列

(2) 以列序为主序，a[i][j]的地址为

$$Loc(i,j)=Loc(0,0)+(m*j+i)*s$$
$$=b+(m*j+i)*s \qquad 0\leqslant i\leqslant m-1,\ 0\leqslant j\leqslant n-1$$

其中： b为首地址，s为每个元素所占的存储单元数

n：列数　　m：行数

例5 三维数组a[1..p, 1..m, 1..n], 假定无0页0行0列

1)以行序为主序，a[k][i][j]的地址为

$$Loc(k,i,j)=Loc(1,1,1)+(m*n*(k-1)+n(i-1)+j-1)*s$$
$$=b+(m*n*(k-1)+n(i-1)+j-1)*s$$
$$1\leqslant k\leqslant p,\quad 1\leqslant i\leqslant m,\quad 1\leqslant j\leqslant n$$

其中：

b为首地址，s为每个元素所占的存储单元数

p---页数 m--行数 n-列数

(2)以列序为主序，a[k][i][j]的地址为

$$Loc(k,i,j)=Loc(1,1,1)+(p*m*(j-1)+p*(i-1)+k-1)*s$$
$$=b+(p*m*(j-1)+p*(i-1)+k-1)*s$$
$$1\leqslant k\leqslant p,\quad 1\leqslant i\leqslant m,\quad 1\leqslant j\leqslant n$$

其中：

b为首地址，s为每个元素所占的存储单元数

p：页数 m：行数 n：列数

例5 三维数组a[0..p-1, 0..m-1, 0..n-1],

(1) 以行序为主序，a[k][i][j]的地址为

$$Loc(k,i,j)=Loc(0,0,0)+(m*n*k+n*i+j)*s$$
$$=b+(m*n*k+n*i+j)*s$$
$$0 \leqslant k \leqslant p-1, \quad 0 \leqslant i \leqslant m-1, \quad 0 \leqslant j \leqslant n-1$$

其中：

b为首地址，s为每个元素所占的存储单元数

p--页数　m--行数　n-列数

(2) 以列序为主序，a[k][i][j]的地址为

$$Loc(k,i,j)=Loc(0,0,0)+(p*m*j+p*i+k)*s$$
$$=b+(p*m*j+p*i+k)*s$$
$$0 \leqslant k \leqslant p-1, \quad 0 \leqslant i \leqslant m-1, \quad 0 \leqslant j \leqslant n-1$$

其中：

b为首地址，s为每个元素所占的存储单元数

p：页数 m：行数　n：列数

华中科技大学

5.3 矩阵的压缩存储

5.3.1特殊矩阵的压缩存储

1. n阶对称矩阵

上三角

$$A_{nxn}=\begin{pmatrix} a_{11} & & a_{1n}\\ & & a_{ji} & \\ & & & \\ a_{i1}\cdots a_{ij} & & \cdots\cdots & \\ a_{n1} & & & a_{nn}\end{pmatrix}$$

下三角

$a_{ij}==a_{ji}$

$1\leqslant i, j\leqslant n$

假定以行序为主，顺序存储下三角元素到SA[1..maxleng]

a_{11}	a_{21}	a_{22}	a_{31}	...	a_{i1}	...	a_{ij}	...	a_{ii}	...	a_{n1}	...	a_{nn}

k=1　2　3　4　...　i(i-1)/2+j　...　n(n+1)/2

如何求a_{ij}在SA中的位置，即序号k？

(1) 设aij在下三角，i≥j

∵ 第1~i-1行共有元素

$1+2+3+\cdots(i-1)=i(i-1)/2$（个）

ai1~aij共有j个元素

∴ aij的序号为：

$k=i(i-1)/2+j$

(2) 设a_{ij}在上三角，$i<j$

∵ 上三角的a_{ij}=下三角的a_{ji}

下三角的a_{ji}的序号为

$k=j(j-1)/2+i \qquad i<j$

∴ 上三角的a_{ij}的序号为

$k=j(j-1)/2+i \qquad i<j$

由(1)和(2)，任意a_{ij}在SA中的序号，为

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

称为在SA中的映象函数，下标转换公式。

2. 三对角矩阵

$$A_{nxn}=\begin{pmatrix} a_{11} & a_{12} & & & & & \text{全}0 \\ a_{21} & a_{22} & a_{23} & & & & \\ & a_{32} & a_{33} & a_{34} & & & \\ & & \ldots & a_{ij} & \ldots\ldots & & \\ \text{全}0 & & & a_{n-1n-2} & a_{n-1n-1} & a_{n-1n} \\ & & & & a_{nn-1} & a_{nn} \end{pmatrix}$$

假定以行序为主，顺序存储非0元素到SA[1..maxleng]：

a_{11}	a_{12}	a_{21}	a_{22}	a_{23}	a_{32}	...	a_{ij}	...	a_{nn-1}	a_{nn}
k=1	2	3	4	5	6	...	?	...		3n-2

任意$a_{ij}\neq 0$，在SA中的序号：$k=(3*(i-1)-1)+(j-i+2)=2i+j-2$

$$A[i,j]=\begin{cases} SA[k] & |i-j|<=1 \\ 0 & \text{其它} \end{cases}$$

5.3.2 稀疏矩阵的压缩存储

1.三元组表

例　稀疏矩阵M及其转置矩阵T

$$
M=\begin{pmatrix}
0 & 13 & 9 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-3 & 0 & 0 & 0 & 0 & 14 & 0 \\
0 & 0 & 24 & 0 & 0 & 0 & 0 \\
0 & 18 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -7 & 0 & 0 & 0
\end{pmatrix}
$$

表B

i	j	e=aij
1	2	10
2	5	12
2	7	8
3	1	19
3	4	28
7	6	7
8	3	66

稀疏矩阵

表A

i	j	e=aij
1	2	13
1	3	9
3	1	-3
3	6	14
4	3	24
5	2	18
6	4	-7

?

M的三元表存储结构

1	2	13
1	3	9
3	1	-3
3	6	14
4	3	24
5	2	18
6	4	-7
///	///	///
行数(mu)：6		
列数(nu)：7		
非零元(tu)：7		

用C语言定义三元组表

```
#define  MAXSIZE 100
typedef struct  {
   int        i,j;//非零元行、列下标
   ElemType  e;
 } Triple;  //定义三元组

typedef struct  {
   Triple  data[MAXSIZE+1];
   int  mu,nu,tu;
 } TSMatrix; //定义三元组表

TSMatrix M;
```

求转置矩阵

$$M=\begin{pmatrix} 0 & 13 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 24 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -7 & 0 & 0 & 0 \end{pmatrix}$$

M的三元表存储结构

1	2	13
1	3	9
3	1	-3
3	6	14
4	3	24
5	2	18
6	4	-7
///	///	///

行数(mu)：6
列数(nu)：7
非零元(tu)：7

$$T=\begin{pmatrix} 0 & 0 & -3 & 0 & 0 & 0 \\ 13 & 0 & 0 & 0 & 18 & 0 \\ 9 & 0 & 0 & 24 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -7 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

T的三元表存储结构

2	1	13
3	1	9
1	3	-3
6	3	14
3	4	24
2	5	18
4	6	-7
///	///	///

行数(mu)：7
列数(nu)：6
非零元(tu)：7

不符合以行为主的存放

M的三元表存储结构

1	2	13
1	3	9
3	1	-3
3	6	14
4	3	24
5	2	18
6	4	-7
///	///	///

行数(mu)：6
列数(nu)：7
非零元(tu)：7

```
T.mu=M.nu;T.nu=M.mu; T.tu=M.tu;
if (T.tu)  {
   q=1;       /*指示向T写时的位置*/
   for(col=1;col<=M.nu; ++col)
      for(p=1;p<=M.tu;++p) /*扫描M三元表*/
         if(M.data[p].j==col) /*当前行*/
               {T.data[q].i=M.data[p].j;
                T.data[q].j=M.data[p].i;
                T.data[q].e=M.data[p].e;
                q++; }
      }
return OK;
```

算法1：
时间复杂度：
O(nu*tu)

col	1	2	3	4	5	6	7
num[col]	1	2	2	1	0	1	0
cpot[col]	1	2	4	6	7	7	8

算法2：
时间复杂度：
O(nu+tu)

cpot[1]=1;

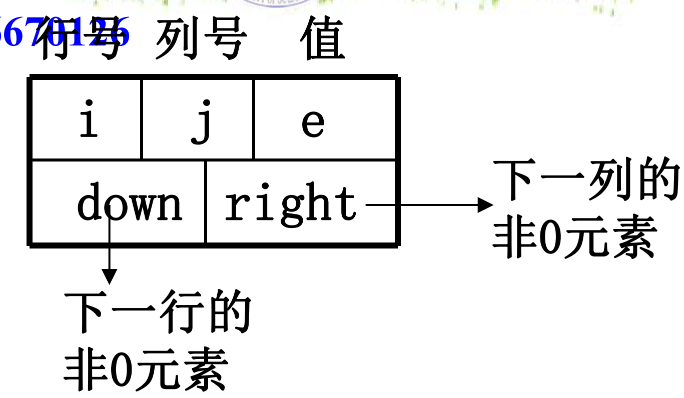
cpot[col]=cpot[col-1]+num[col-1]　　　2≤col≤a.nu

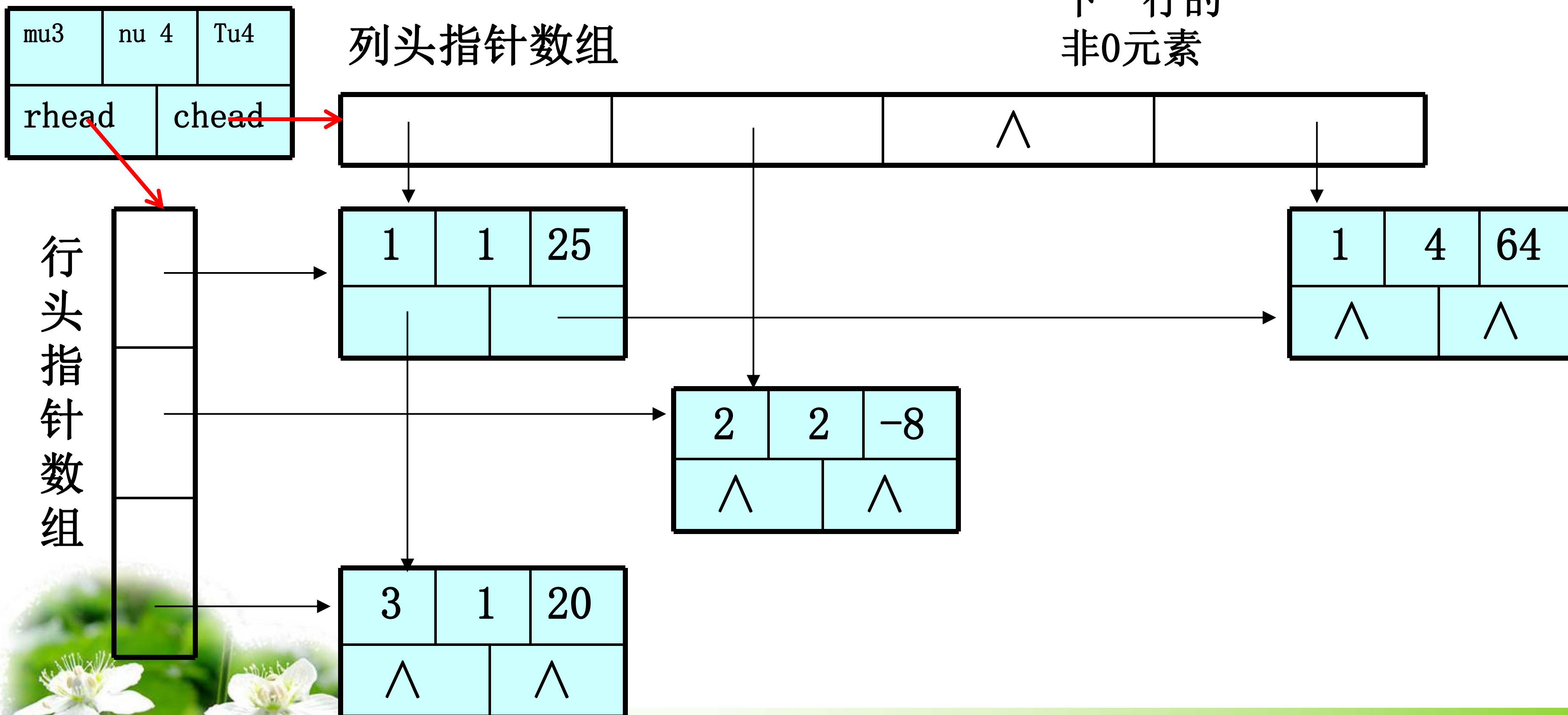
```
for(p=1;p<=M.tu;++p)    //扫描M三元表
 {col=M.data[p].j;      //确定当前元素列号
  q=cpot[col];          //确定当前元素M.data[p]
                          在T的当前存放位置*/
  T.data[q].j=M.data[p].i; T.data[q].i=M.data[p].j;
  T.data[q].e=M.data[p].e;
  ++cpot[col];         /*T的当前列指示下一空位置*/
 }
```

2. 十字链接表

例 稀疏矩阵

$$M=\begin{pmatrix} 25 & 0 & 0 & 64 \\ 0 & -8 & 0 & 0 \\ 20 & 0 & 0 & 0 \end{pmatrix}$$





5.4 广义表(generalized list)，列表(lists)

5.4.1广义表的定义和术语

n(n≥0)个数据元素或广义表的一个有限序列叫做广义表。

记作：$LS=(e_1, e_2, \ldots, e_n)$。 n为LS的长度。

其中：LS----广义表名

e_i----单元素、原子，约定用小写， $1 \le i \le n$

e_i----广义表，约定用大写， $1 \le i \le n$

(1)空表 LS=( )，n=0

(2)非空表 $LS=(e_1, e_2, \ldots, e_n)$ n>0

其中：

e_1---LS的表头/首部，记作：$Head(LS)=e_1$

$(e_2, \ldots, e_n)$--- LS的表尾/尾部，记作：

$Tail(LS)=(e_2, \ldots, e_n)$

华中科技大学

广义表举例:

(1) A=( )　　　// 空表

(2) B=(e)

　　Head(B)=e　　　Tail(B)=( )

(3) C=(a, b, c)

　　Head(C)=a　　　Tail(C)=(b, c)

　　Head(Tail(C))=b　　Tail(Tail(C))=(c)

(4) D=(a, (b, c))

　　Head(D)=a　　　Tail(D)=((b, c))

　　D2=((a, b), c)

　　Head(D2)=(a, b)　　Tail(D2)=(c)

(5) E=((a, b), c, (d, e))

　　Head(E)=(a, b)　　Tail(E)=(c, (d, e))

　　Head(Tail(E))=c　　Tail(Tail(E))=((d, e))

(6) F=(A, B, C, d)=(( ), (e), (a, b, c), d)

Head(F)=( )

Tail(F)=((e), (a, b, c), d)

(7) G=(a, G) //递归广义表

=(a, (a, G))=(a, (a, (a, G)))

=(a, (a, (a, (a, ... G))))

Head(G)=a

Tail(G)=(G)=((a, G))

(8) H=(( ), (( ), ( )))

Head(H)=( )

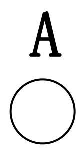
Tail(H)=((( ), ( )))

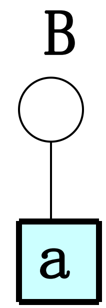
5.4.2 广义表的图型表示----树型结构

约定　□----单元素/原子

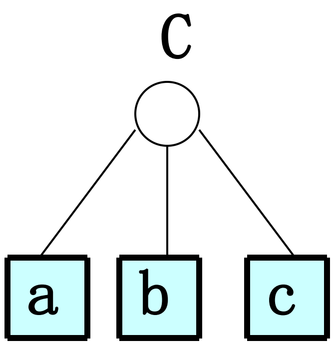
　　　○----列表,若有表名,附表名

例　(1) A=( )　　　　　　　　(2) B=(a)

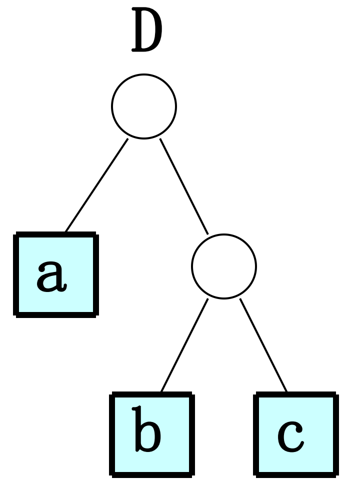




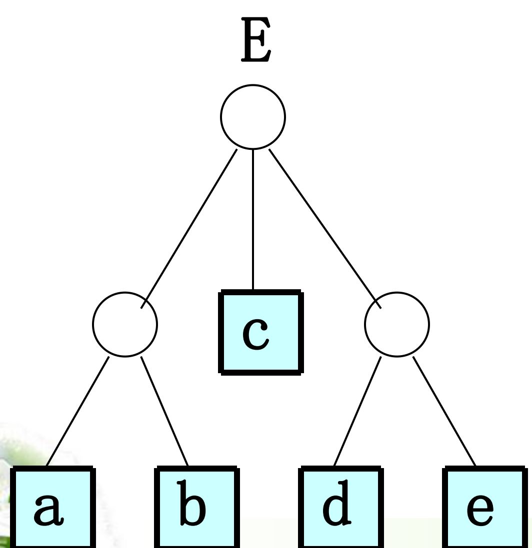
(3) C=(a,b,c)



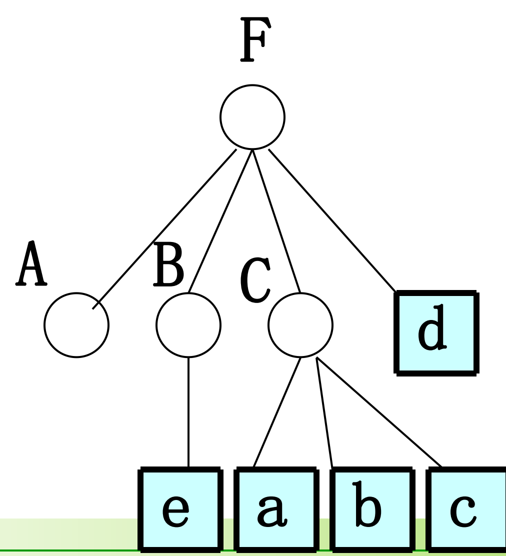
(4) D=(a, (b,c))



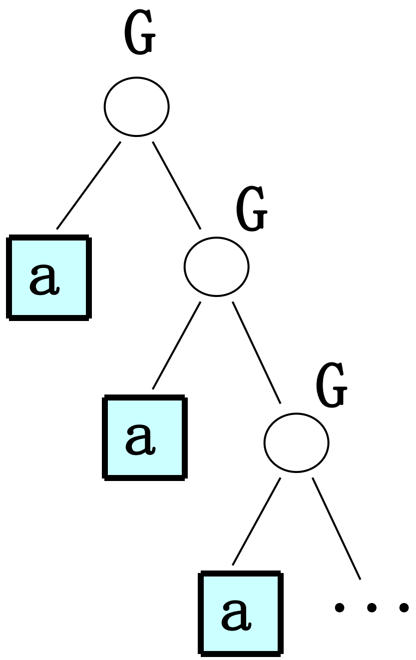
(5) E=((a, b), c, (d, e))



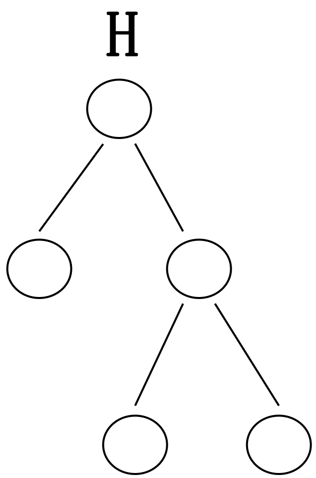
(6) F=(A, B, C, d)=(( ), (e), (a, b, c), d)



(7) G=(a, G)



(8) H=(( ), (( ), ( )))



5.4.3 广义表的操作

1. 求长度：Leng(LS)

a=( )	Leng(A)=0
G=(a, G)	Leng(G)=2
H=(( ), (( ), ( )))	Leng(H)=2
F=(A, B, C, d)	Leng(F)=4

2. 求表头： Head(LS)

G=(a, G)	Head(G)=a
E=((a, b), c, (d, e))	Head(E)=(a, b)

3. 求表尾： Tail(LS)

G=(a, G)　　Tail(G)=(G)=((a, G))
E=((a, b), c, (d, e))　　Tail(E)=(c, (d, e))

4. 求第i个元素：GetElem(LS, i)=ei　　1≤i≤n

I=((a, b), c, ( ), (d))
GetElem(I, 1)=(a, b)　　Get(I, 2)=C
GetElem(E, 3)=( )　　Get(I, 4)=(d)

5. 求深度： Depth(LS)---LS所含括号的层数

(1) A=( )　　　　　　　　Depth(A)=1

(2) E=((a, b), c, (d, e))　　Depth(E)=2

(3) H=(( ), (( ), ( )))　　Depth(H)=3

6. 插入：InsertFirst(LS, e)---e插入LS的第一个位置

设　A=( )

执行：InsertFirst(A, a)；

得：　A=(a)

执行：InsertFirst(A, (b, (c)))；

得：　A=((b, (c)), a)

执行：InsertFirst(A, ( ))；

得：　A=(( ), (b, (c)), a)

7. 其它：　......

5.5 广义表的存储结构

广义表的的元素具有不同结构，一般用链式存储结构。

原子结点：

tag=0	atom（元素）

列表结点：

tag=1	hp（表头）	tp（表尾）

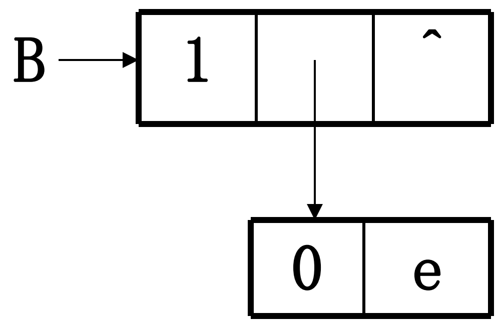
```
typedef  struct GLNode{
    ElemTag  tag;
    union { AtomType atom;
            struct { struct GLNode    *hp, *tp; }ptr;
          }
     } *GList;
```

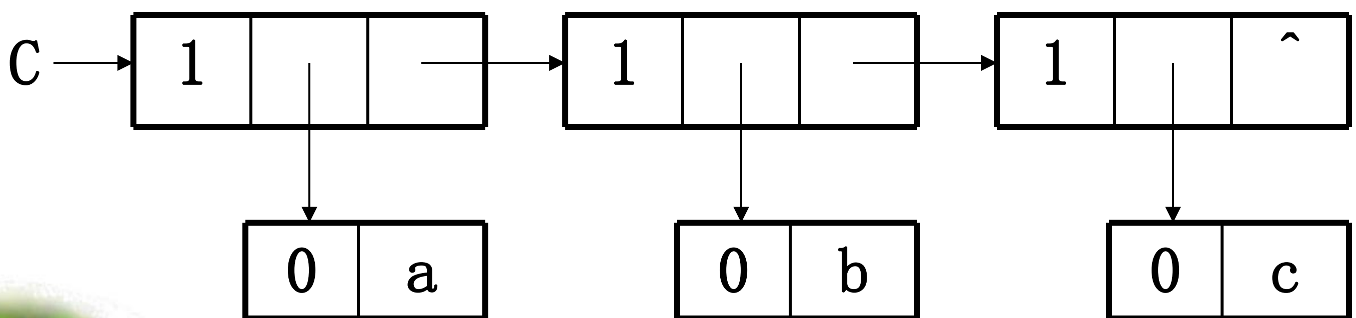
(1) A=( )

(2) B=(e)

(3) C=(a, b, c)

A=NULL

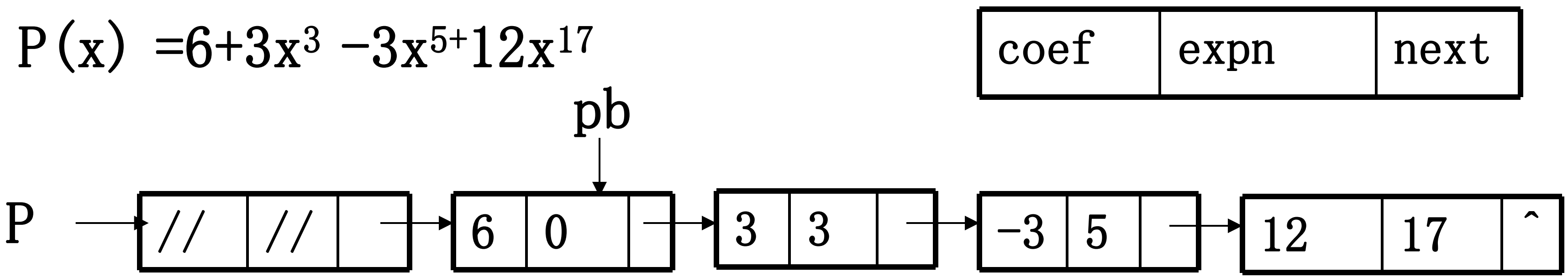




5. 6** m元多项式的表示

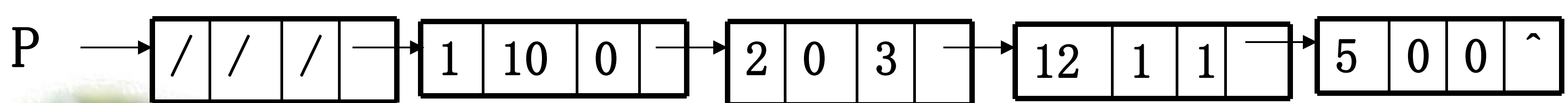
一元多项式：

P(x) =6+3x³ −3x⁵+12x¹⁷



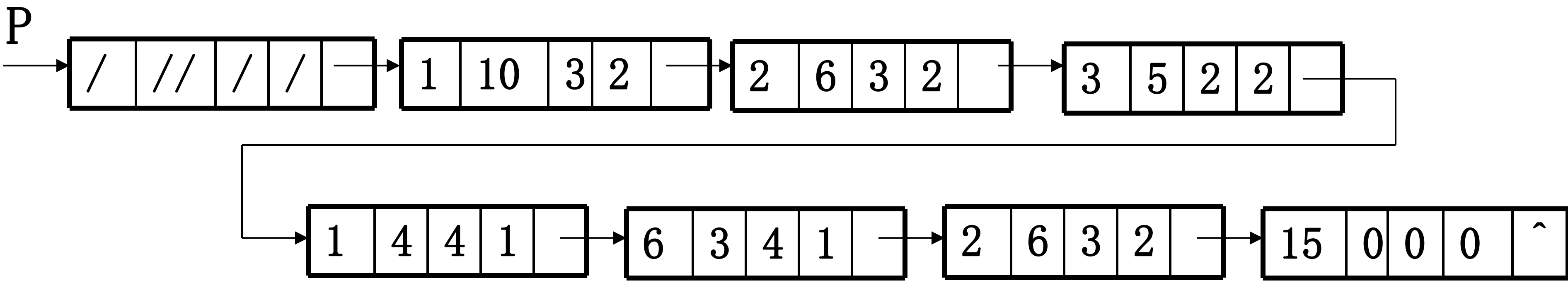
二元多项式：

P(x, y) =x¹⁰ + 2y³ +12yz+ 5



三元多项式：

$P(x, y, z) = x^{10}y^3z^2 + 2x^6y^3z^2 + 3x^5y^2z^2 + x^4y^4z + 6x^3y^4z + 2yz + 15$



缺点：

➢若m元多项式无论各项的变元数多少，都按m个指数分配单元，造成空间浪费，若按实际分配，则操作困难；

➢m值不同，结点大小不一致，存储管理

华中科技大学

三元多项式变形:

$P(x,y,z) = ((x^{10} + 2x^{6})\, y^{3} + 3\, x^{5}y^{2})\, z^{2} + ((x^{4} + 6x^{3})\, y^{4} + 2y)\, z + 15$

$A(x,y) = (x^{10} + 2x^{6})\, y^{3} + 3\, x^{5}y^{2}$

$B(x,y) = (x^{4} + 6x^{3})\, y^{4} + 2y$

P=z((A,2), (B,1) , (15,0))

$C(x) = x^{10} + 2x^{6}$

$D(x) = 3\, x^{5}$

$E(x) = x^{4} + 6x^{3}$

$F(x) = 2$

A=y((C,3), (D,2))

B=y((E,4) , (F,1))

C=x((1,10), (2,6))

D=x((3,5))

E=x((1,4), (6,3))

F=x((2,0))

原子结点：

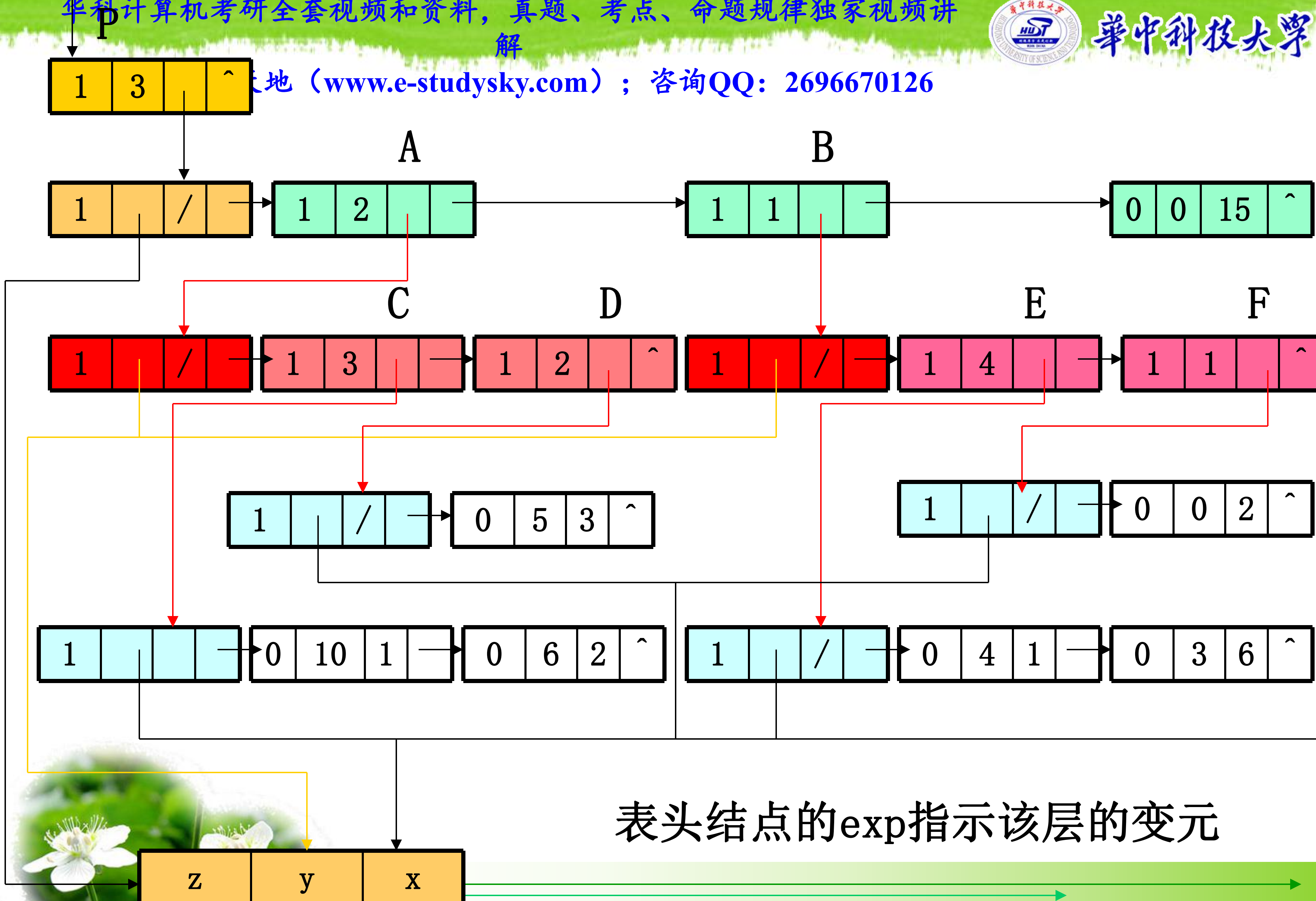
tag=0	exp	coef	tp

列表结点：

tag=1	exp	hp	tp

其中：exp为指数域；coef为系数域，tp指向同层下一结点

```
typedef  struct MPNode{
    ElemTag  tag;
    int       exp;                  //指数域
    union { float coef;           //系数域
            struct MPNode  *hp;//表结点的指针域
          };
    struct MPNode  *tp;         //指向同层下一结点
          } *MPList;
```



表头结点的exp指示该层的变元

课堂练习

1。设n维数组$A[l_1\cdots u_1]\cdots[l_n\cdots u_n]$按行序进行存放数据元素，写出数组元素$A[i_1][i_2]\cdots[i_n]$的寻址公式

2 如下图所示特殊矩阵，设计压缩存储方案。

$$
\begin{bmatrix}
 & & & & & & a_{1,2m-1} & a_{1,2m} \\
 & & & & & & a_{2,2m-1} & a_{2,2m} \\
 & & 0 & & a_{3,2m-3} & a_{3,2m-2} & & \\
 & & & & a_{4,2m-3} & a_{4,2m-2} & & \\
 & & & & \cdots & & & \\
 & & & a_{i,j} & & & & \\
 & & \cdots & & & & 0 & \\
a_{2m-1,1} & a_{2m-1,2} & & & & & & \\
a_{2m,1} & a_{2m,2} & & & & & &
\end{bmatrix}
$$