

三元组顺序表，稀疏矩阵的三元组表示及（C语言）实现

本节介绍稀疏矩阵的三元组[顺序表](#)压缩存储方式。

通过《[矩阵的压缩存储](#)》一节我们知道，稀疏矩阵的压缩存储，至少需要存储以下信息：

- 矩阵中各非 0 元素的值，以及所在矩阵中的行标和列标；
- 矩阵的总行数和总列数；

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 5 \\ 3 & 0 & 0 \end{pmatrix}$$

图 1 稀疏矩阵示意图

例如，图 1 是一个稀疏矩阵，若对其进行压缩存储，矩阵中各非 0 元素的存储状态如图 2 所示：

数组	(1 , 1 , 1)	(2 , 3 , 5)	(3 , 1 , 3)
	0	1	2

图 2 稀疏矩阵的压缩存储示意图

图 2 的[数组](#)中，存储的是三元组（即由 3 部分数据组成的集合），组中数据分别表示（行标，列标，元素值）。

注意，这里矩阵的行标和列标都从 1 开始。

C 语言中，三元组需要用结构体实现，如下所示：

```
01.  //三元组结构体
02.  typedef struct {
03.      int i,j;//行标i, 列标j
04.      int data;//元素值
05.  }triple;
```

由于稀疏矩阵中非 0 元素有多个，因此需要建立 triple 数组存储各个元素的三元组。除此之外，考虑到还要存储矩阵的总行数和总列数，因此可以采用以下结构表示整个稀疏矩阵：

```

01. #define number 20
02. //矩阵的结构表示
03. typedef struct {
04.     triple data[number]; //存储该矩阵中所有非0元素的三元组
05.     int n,m,num; //n和m分别记录矩阵的行数和列数, num记录矩阵中所有的非0元素的个数
06. }TSMatrix;

```

可以看到, TSMatrix 是一个结构体, 其包含一个三元组数组, 以及用于存储矩阵总行数、总列数和非 0 元素个数的变量。

假设采用 TSMatrix 结构体存储图 1 中的稀疏矩阵, 其 C 语言实现代码应该为:

```

01. #include<stdio.h>
02. #define number 3
03. typedef struct {
04.     int i,j;
05.     int data;
06. }triple;
07. typedef struct {
08.     triple data[number];
09.     int n,m,num;
10. }TSMatrix;
11. //输出存储的稀疏矩阵
12. void display(TSMatrix M);
13. int main() {
14.     TSMatrix M;
15.     M.m=3;
16.     M.n=3;
17.     M.num=3;
18.
19.     M.data[0].i=1;
20.     M.data[0].j=1;
21.     M.data[0].data=1;
22.
23.     M.data[1].i=2;
24.     M.data[1].j=3;
25.     M.data[1].data=5;
26.
27.     M.data[2].i=3;
28.     M.data[2].j=1;
29.     M.data[2].data=3;
30.     display(M);
31.     return 0;
32. }
33. void display(TSMatrix M){
34.     for(int i=1;i<=M.n;i++){

```

```
35.         for(int j=1;j<=M.m;j++){
36.             int value =0;
37.             for(int k=0;k<M.num;k++){
38.                 if(i == M.data[k].i && j == M.data[k].j){
39.                     printf("%d ",M.data[k].data);
40.                     value =1;
41.                     break;
42.                 }
43.             }
44.             if(value == 0)
45.                 printf("0 ");
46.         }
47.         printf("\n");
48.     }
49. }
```

输出结果为：

```
1 0 0
0 0 5
3 0 0
```

[< 上一节](#)

[下一节 >](#)

[联系方式](#) [购买教程（带答疑）](#)