

5.1 稀疏矩阵快速转置

稀疏矩阵指零元较多的矩阵。对于稀疏矩阵最有效的存储方法是只存储非零元，但这也失去了按下标随机访问的特性，使求解问题的方法发生根本的改变。例如矩阵转置算法不能直接套用到稀疏矩阵的转置中。

一、程序设计简介

本验证程序只有一个源程序文件 `MatrixTrans.cpp`，实现稀疏矩阵的转置。稀疏矩阵采用了三元组的顺序存储方式，矩阵元素为整型。为方便操作，设计了用户通过一个二维数组给出非零元信息的方法，如：

```
int da[6][3]={{0,2,11},{0,4,12},{1,3,22},{2,1,31},{2,3,32},{3,0,41}};
```

程序中的函数 `Mcreate()`根据上述二维数组生成三元组顺序存储的稀疏矩阵。为了增加直观性，显示时给出了包括零元的完整矩阵，由函数 `Mdisp()`实现。函数 `MatrixTrans_1()`实现转置算法一（算法 4.1），`MatrixTrans_2()`实现快速转置（算法 4.2）。函数调用关系如图 1.6.1 示。

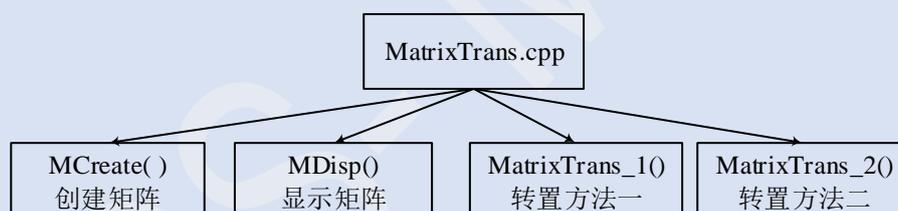


图 1.6.1 稀疏矩阵转置函数调用关系

二、运行说明

编译与链接成功后，屏幕显示转置矩阵和两种方法转置后的矩阵。程序运行结果如图 1.5.2 所示。

```

E:\数据结构实践教程V2-源码\4_数组和矩阵\4...
ma=
  0   0   11   0   12   0
  0   0   0   22   0   0
  0  31   0  32   0   0
 41   0   0   0   0   0

方法一：直接取，顺序存
mb=
  0   0   0   41
  0   0  31   0
 11   0   0   0
  0  22  32   0
 12   0   0   0
  0   0   0   0

方法二：顺序取，直接存
mb=
  0   0   0   41
  0   0  31   0
 11   0   0   0
  0  22  32   0

```

图 1.5.2 矩阵快速转置运行结果示意

三、思考题

1. 阅读源程序，回答下列问题。

- (1) 如何修改被转置矩阵？
- (2) 本程序采用了以行为主的存储方法还是以列为主存储方法？
- (3) 函数 `TSMatrix MCreate(int d[][3],int m,int n,int k)` 中的“3”表示什么意思？为什么是“3”？
- (4) 转置方法一的时间复杂度是多少？
- (5) 快速转置的时间复杂度是多少？
- (6) 三元组非零元的存储空间 `M.data[]` 采用了动态申请，这样做有什么好处？
- (7) 矩阵显示函数 `MDisp(TSMatrix a)` 的时间复杂度是多少？

2. 运行程序，回答下列问题：

- (8) 用下列矩阵运行程序，验证结果的正确。

$$\begin{pmatrix} 2 & 0 & 0 & 12 & 0 & 5 \\ 0 & 9 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 22 & 0 & 0 & 0 & 0 \end{pmatrix}$$