

## Spec. for Proejct II

从以下算法中选择一个实现：

### 1. 稀疏矩阵乘法

设计一个程序实现稀疏矩阵乘法。

**输入：**采用列压缩方式存储的两个稀疏矩阵 A、B

**输出：**采用列压缩方式存储的稀疏矩阵  $C=A*B$

**程序框架：**

提供了稀疏矩阵加法和减法的算法供参考(sadd.c, ssub.c)，这些算法都在 Matlab 2008b 环境下采用 Mex function 实现，在 Matlab 下，可以通过 mex sadd.c 或者 mex ssub.c 来进行编译，编译后可以直接通过  $C = \text{sadd}(A,B)$ 或  $C=\text{ssub}(A,B)$ 来调用这两个程序。稀疏矩阵的乘法也提供了一个 mex 函数的框架，只需要实现 smul() 函数即可，实现之后仍可以通过 mex 命令进行编译。

**关于程序测试：**

在 Matlab 下可以通过  $A = \text{sprand}(\text{rows}, \text{cols}, \text{density})$ ;  $B = \text{sprand}(\text{rows}, \text{cols}, \text{density})$ ; 来随机构造稀疏矩阵。可以通过  $C1 = \text{smul}(A,B)$ ;  $C2=A*B$ ;  $\text{err}=\text{ssub}(C1,C2)$ ;来判断计算结果是否与 Matlab 结果一致。

### 2. 基于 Minimum Degree 算法的稀疏矩阵 Reordering 算法

设计一个程序实现基于 Minimum Degree 算法的稀疏矩阵 Reordering 算法。

**输入：**采用列压缩方式存储的对称稀疏矩阵 A

**输出：**基于 Minimum Degree 算法的系数矩阵 A 的 Permutation vector

**程序框架：**

提供了程序框架 smd.c，只需要实现 md\_reordering()函数即可。在 matlab 中可以通过 mex smd.c 来编译程序，并通过  $p = \text{smd}(A)$ 来调用程序。

**程序测试：**

提供 A1.mat 以及 A2.mat 两个例子供测试使用。比较 Reordering 前后 LU 分解后的矩阵非零元个数。代码如下：

```
[L,U]=lu(A); nnz1=nnz(L+U);[L,U]=lu(A(p,p));[L,U]=lu(A);nnz2=nnz(L+U);
reduce=nnz1 - nnz2;
```