# Parallel Algorithms HW 1

Sparse matrix linear algebra will be one of our main topics. Real world matrices $A \in \mathbb{R}^{n \times n}$ are frequently large ($(n > 10^6)$, sparse (99% or more zero) and unstructured. We want to get some experience working with such matrices.

1. look at the real world matrices at http://math.nist.gov/MatrixMarket/ and download three different matrices. Import them into the software of your choice and recreate the structure plot from MatrixMarket for these matrices. Report the name, a brief description of the application, size, sparsity, and memory used for these matrices.
2. Matrix Multiply
   - **2.1.** Create suitable vectors for your matrices and compute $A.x$. Report the time it took to do the matrix vector multiply in seconds. Note you may need to average the timings to get a good measurement.
   - **2.2.** Convert the matrices to dense storage and report the memory used and the timings for the mutpiplication in dense storage format.
   - **2.3.** Plot the times against matrix size on a (log-log scale) for the dense and full storage.
3. Matrix Multiply
   - **3.1.** Create suitable right hand sides for your matrices and solve the resulting linear systems. Check the accuracy of your results and report the time it took to solve the linear system in seconds. Note you may need to average the timings to get a good measurement.
   - **3.2.** Convert the matrices to dense storage and report the memory used and the timings to solve the linear systems in dense storage format.
   - **3.3.** Plot the times against matrix size on a (log-log scale) for the dense and full storage.
4. Download at least six different sized problems from one problem family in MatrixMarket. Repeat the sparse and dense timing exercises and plot the memory used against matrix size and the solution time against matrix size (both on log-log scales) for the family. Do both the multiplication and linear solve.