1 Submission Requirement

1. Prepare a report including
   - detailed answers to each question
   - numerical results and their interpretation

2. The programming language can be either matlab, Python or c/c++.

3. Pack all of your codes named as "proj1mk-name-ID.zip" send it to TA: pkuopt@163.com

4. 请勿大量将代码粘在报告中，涉及到实际结果需要打表或者作图，不要截图或者直接从命令行拷贝结果。

5. 提交word的同学需要提供word原文件并将其转换成pdf文件。

6. If you get significant help from others on one routine, write down the source of references at the beginning of this routine.

2 Algorithms for $\ell_1$ minimization

Consider the problem

\[
\min_x \mu \|x\|_1 + \|Ax - b\|_1,
\]

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ are given. Test data are as follows:

\[
\begin{align*}
  n &= 1024; \\
  m &= 512; \\
  A &= \text{randn}(m,n); \\
  u &= \text{sprandn}(n,1,0.1); \\
  b &= A*u; \\
  \mu &= 1e-2;
\end{align*}
\]

See [http://bicmr.pku.edu.cn/~wenzw/bigdata/Test_BP.m](http://bicmr.pku.edu.cn/~wenzw/bigdata/Test_BP.m)
1. Solve (2.1) using CVX by calling different solvers mosek or gurobi.
   CVX, Mosek and Gurobi are available free at:

2. Write down and implement one of the following algorithms in Matlab/Python:
   (a) Classical Augmented Lagrangian method (or Bregman method), where each augmented Lagrangian function is minimized by using the proximal gradient method
       Reference: Wotao Yin, Stanley Osher, Donald Goldfarb, Jerome Darbon, *Bregman Iterative Algorithms for l1-Minimization with Applications to Compressed Sensing*
   (b) Classical Augmented Lagrangian method (or Bregman method), where each augmented Lagrangian function is minimized by using the accelerated proximal gradient method (FISTA or Nesterov’s method)
       Reference on FISTA: Amir Beck and Marc Teboulle, *A fast iterative shrinkage thresholding algorithm for linear inverse problems*

3. Write down and implement one of the following algorithms in Matlab/Python:
   (a) Alternating direction method of multipliers (ADMM) for the primal or dual problem
   (b) Alternating direction method of multipliers with linearization for the primal or dual problem

4. (Optional) Develop algorithms for solving the following problems:
   \[
   \begin{align*}
   \min_x & \quad \mu \|x\|_1 + \|Ax - b\|_2, \\
   \min_x & \quad \mu \|x\|_1 + \|Ax - b\|_\infty, \\
   \min_x & \quad \mu \|x\|_{1/2} + \|Ax - b\|_2,
   \end{align*}
   \]
   where \(\|x\|_{1/2} = \sum_i |x_i|^{1/2}\).

5. Requirement:
   (a) The interface of each method should be written in the following format
       
       \[ [x, out] = \text{method_name}(x0, A, b, mu, opts); \]
       Here, \(x0\) is a given input initial solution, \(A\) and \(b\) are given data, \(opts\) is a struct which stores the options of the algorithm, \(out\) is a struct which saves all other output information.
   (b) Compare the efficiency (cpu time) and accuracy (checking optimality condition) in the format as
       [http://bicmr.pku.edu.cn/~wenzw/bigdata/Test_BP.m](http://bicmr.pku.edu.cn/~wenzw/bigdata/Test_BP.m)
3 Algorithms For Low-rank Recovery

Consider the model

\[
\min_{X \in \mathbb{R}^{m \times n}} \mu \|X\|_* + \sum_{(i,j) \in \Omega} (X_{ij} - M_{ij})^2,
\]

where the nuclear norm \( \|X\|_* = \sum \sigma_i(X) \).

1. Write down and implement a proximal gradient method for solving (3.1).
2. Write down and implement an alternating direction method of multipliers (ADMM) for solving (3.1).
3. The data \( M \) and \( \Omega \) are specified in the following script:
   
   \texttt{http://bicmr.pku.edu.cn/~wenzw/bigdata/Test_MC.m}

   Test your method for \( \mu = 10^{-1}, 10^{-2}, 10^{-3} \).
4. (Optional) Design a method for solving the following problem:

\[
\min_{X \in \mathbb{R}^{m \times n}} \mu \|X\|_* + \sum_{(i,j) \in \Omega} |X_{ij} - M_{ij}|.
\]