Some of the question below have instruction in **bold** that require you to make plots or answer questions. Create a short pdf document called "hmwk3.pdf" with your results for each question. You don't need to make an elaborate writeup, just put your plots into the pdf with labels indicating what they are, and answer questions when prompted.

1. (a) (matlab users only) Download the solver FASTA (http://www.cs.umd.edu/ tomg/FASTA.html). Read the first 2 pages of the users manual. This solver performs general forward backward splitting with a user-supplied gradient and proximal operator.

   (b) (python users only) Write a method that uses FBS to solve a general problem of the form

   $$\text{minimize} \quad f(x) + g(x).$$

   You can do this by modifying your `grad_descent` code from homework 2. Your new method should have signature

   ```
   def fbs(f, gradf, proxg, x0):
           ...
   ```

   The argument `f` computes the scalar-valued function $f(x)$. The argument `gradf` is a function handle that computes the gradient of f. This means that

   $$\texttt{gradf}(x) = \nabla f(x).$$

   The argument `proxg` is a function handle that computes the proximal operator of g with stepsize τ. This means that

   $$\texttt{proxg}(z, \tau) = \arg\min_x g(x) + \frac{1}{2\tau}\|x - z\|^2$$

   Your method should start by estimating the initial stepsize τ using the Lipschitz constant for the gradient of f. You already did this in homework 2 in your `grad_descent` method. The method should then perform an iteration of FBS, and check that the following backtracking condition holds:

   $$f(x^{k+1}) \le f(x^k) + \langle x^{k+1} - x^k, \nabla f(x^k)\rangle + \frac{1}{2\tau}\|x^k - x^{k+1}\|^2.$$

   If this condition fails, then throw away x^{k+1}, let $\tau \leftarrow \tau/2$, and redo the iteration starting at x^k. Your method should terminate when the simplified residual $\frac{1}{\tau}\|x^{k+1} - x^k\|$ is "small."

2. Use the FASTA code (or your python solver) to solve the sparse logistic regression problem

   $$\text{minimize} \quad \mu|x| + \text{logreg\_objective}(x, \hat{D}, c)$$

   where `logreg_objective` is the method you wrote in homework 1. Build the classification problem by choosing $[D, c] = \text{create\_classification\_problem}(200, 5, 1)$, and then append a bunch of random columns to D to form $\hat{D} = [D|G]$ where G is a matrix of 5 random Gaussian columns. Pick some reasonable value of μ that generates a sparse solution, and solve the problem using FBS. To do this, you will need to call your FBS solver with $g(x) = \mu|x|$.

   Plot your (sparse) solution, and plot the residual as a function of iteration number.

3. Use the FASTA code (or your python solver) to solve the problem

$$\text{minimize} \quad \mu|\nabla_d x| + \frac{1}{2}\|x - f\|^2$$

where f is a 2D noisy image and ∇_d is the discrete gradient operator you implemented in homework 1. You can do this using the dual formulation

$$\underset{\lambda}{\text{minimize}} \quad \frac{1}{2}\|\nabla_d^T \lambda - f\|^2 \ \text{ subject to } \ -\mu \leq \lambda_i \leq \mu.$$

Note that λ needs to be a 3D array so that your divergence operator can be applied. Once the problem is solved, you can recover the denoised image as

$$x = f - \nabla^T \lambda.$$

Test your code on an arbitrary noisy test image (for example the Shepp-Logan phantom, or a simple synthetic image with a white square in a black background). **Plot your noisy test image, the denoised image, and the residuals.**

4. Build two random binary 50×3 matrices, A and B. Half the entries should be 1 and others are 0. Compute $D = AB^T$. Now, write a code that tries to recover the original matrices by solving the convex relaxation

$$\underset{X,Y}{\text{minimize}} \quad \frac{1}{2}\|XY - D\|^2 \ \text{ subject to } \ 0 \leq X_{i,j}, Y_{i,j} \leq 1$$

using FBS. **Plot the residuals as a function of iteration number. Does the recovery method work? Does it depend on the initialization?**