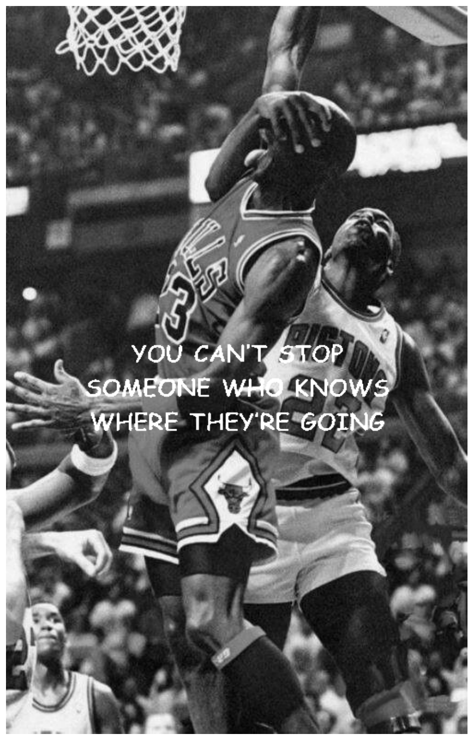
ORF 363/COS 323	Problem set 6	Fall 2025, Princeton University
Instructor: A.A. Ahmadi		
TAs: Budway, Hua, Shi, Tang, Yang		
Due on Thursday, November 13, 2025, at 1:30pm EST, on Gradescope		

For all problems that involve coding, please include your code.

Problem 1: You can't stop someone who knows where they are going



You love the image of Michael Jordan that you see here and want to set it as your desktop background. However, you think the text makes the message too obvious, and you would rather keep it more subtle. Therefore, you wish to restore the image to its original, text-free version. The image is grayscale and is given by the file `MJ.mat`. You can load this file to MATLAB by running

```
load('MJ.mat')
V = double(MJ)
```

or to Python by running

```
from scipy.io import loadmat
V = loadmat('MJ.mat')['MJ']
```

The matrix $V \in \mathbb{R}^{m\times n}$, with $m = 790, n = 500$, has entries in $[0, 1]$, where 0 representing pure black, 1 representing pure white, and numbers in between representing different shades of gray. The text pixels are the only pixels in pure white. To automatically remove the text, consider solving the following optimization problem:

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}} \quad & f(X) := \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \left\| \begin{bmatrix} X_{i+1,j} - X_{ij} \\ X_{i,j+1} - X_{ij} \end{bmatrix} \right\|_1 \\ \text{s.t.} \quad & X_{ij} = V_{ij} \quad \text{if } V_{ij} \neq 1, \end{aligned} \tag{1}$$

where $\|\cdot\|_1$ denotes the 1-norm. The constraints ensure that we leave the pixels that are not related to text unchanged. The objective is to minimize the deviation of the remaining pixel values (i.e., the ones corresponding to text) from their neighbors.

(a) Show that problem (1) is a linear program (you don't need to write it in standard form).

(b) Solve problem (1), report the optimal value, and display your optimal matrix. Assuming your decision matrix is called X, you can display your image in MATLAB using the command `imshow(X)`, and in Python by running `import matplotlib.pyplot as plt` and `plt.imshow(X.value, cmap='gray')`.

Implementation hint: For an efficient implementation of this problem, code up the objective function in the following equivalent form

$$f(X) = g(X_{2:m,1:n-1} - X_{1:m-1,1:n-1}) + g(X_{1:m-1,2:n} - X_{1:m-1,1:n-1}).$$

Here, the notation $X_{2:m,1:n}$ for example refers to a submatrix of X which has rows 2 through m and columns 1 through n of X (both ends inclusive). For a general matrix variable A, the function g is defined as $g(A) := \sum_i \sum_j |A_{ij}|$ and can be implemented in MATLAB as `sum(sum(abs(A)))` and in Python as `cp.sum(cp.abs(A))`.

To efficiently implement the constraints of (1), consider defining an $m \times n$ indicator matrix which has $(i, j)^{\text{th}}$ element equal to 1 if $V_{ij} \neq 1$ and zero otherwise. Then consider entrywise multiplying this matrix with V and with your decision matrix.

Problem 2: Deciphering a secret

One of your classmates (no names mentioned) competes with ChatGPT in hallucination and believes LeBron James is better than Michael Jordan. AAA has a secret wake-up-call message for him, which he first writes in binary as a vector $x^{secret} \in \{0,1\}^{304}$. He then encrypts this binary message by generating a random matrix $A \in \mathbb{R}^{170\times 304}$ (using the command `A=randn(170,304)` in MATLAB) and then computing a vector $y \in \mathbb{R}^{170}$ as $y = Ax^{secret}$. (Note that the vector y here is much smaller in length than x^{secret}, so a priori one would think that y has lost much of the information contained in x^{secret}!) AAA then shares A and y in the file `encrypted_secret.mat`, hoping that his students would be able to recover the original message x^{secret} using their optimization knowledge.

To load A and y in MATLAB, you can run

```
load('encrypted_secret.mat')
```

In Python, you can run

```
from scipy.io import loadmat
data = loadmat('encrypted_secret.mat')
A = np.array(data['A'], dtype=float)
y = np.array(data['y'], dtype=float).reshape(170)
```

(a) To recover x^{secret}, let $n = 304$ and consider the optimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \sum_{i=1}^{n}(x_i - x_i^2) \\ \text{s.t.} \quad & Ax = y \\ & 0 \le x \le 1. \end{aligned} \tag{2}$$

What is the optimal value of (2)? Justify. Is (2) a convex optimization problem? Justify.

(b) Because problem (2) seems difficult to solve, we replace the objective function with its first-order Taylor approximation at the origin, ending up with the problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \sum_{i=1}^{n} x_i \\ \text{s.t.} \quad & Ax = y \\ & 0 \le x \le 1. \end{aligned} \tag{3}$$

Is (3) a convex optimization problem? Does your optimal solution to (3) allow you to recover an optimal solution to (2)?

(c) Using a binary-to-text converter,[1] decode AAA's wake-up-call message to the student (and tell us what it is).

[1] Available e.g. at `https://codebeautify.org/binary-to-text`.