CS 111 (S19): Homework 2

Due by 6:00pm Monday, April 14

NAME and PERM ID No.: Gaucho Olé, 1234567 (replace with yours)

UCSB EMAIL: GauchoOle@ucsb.edu (replace with yours)

1. Write the following matrix in the form $A = LU$, where L is a unit lower triangular matrix (that is, a lower triangular matrix with ones on the diagonal) and U is an upper triangular matrix. You can check your answer using Python, but for this exercise, you need to also show the steps you took to get to your answer with some explanation to go with them.

$$A = \begin{pmatrix} 5 & -1 & -1 \\ -1 & 5 & -1 \\ -1 & -1 & 5 \end{pmatrix}$$

2. The following three statements are all **false**. For each one, give a counterexample consisting of a 3-by-3 matrix or matrices, and show the computation that proves that the statement fails.

2a. If P is a permutation matrix and A is any matrix, then $PA = AP$.

2b. If matrix A is nonsingular, then it has a factorization $A = LU$ where L is lower triangular and U is upper triangular.

2c. The product of two symmetric matrices is a symmetric matrix.

3a. Consider the permutation matrix

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Find a 4-element permutation vector `v = np.array(something)` such that, for *every* 4-by-4 matrix A, we have `A[v,:] == P @ A`. Test your answer by running a few lines of Python, and turn in the result.

3b. For the same P, find a 4-element permutation vector `w = np.array(something)` such that, for *every* 4-by-4 matrix A, we have `A[:,w] == A @ P`. Test your answer and turn in the result.

4. Write `Usolve()`, analogous to `Lsolve()` that we covered in class in lecture 4 in week 2 (look in the appropriate online lecture file). You are writing this function in order to solve an upper triangular system $Ux = y$. The file you'll be working on and turning in is `LUsolve.py` (skeleton found in the lecture file repository).

Warning: Notice that, unlike in `Lsolve()`, the diagonal elements of U don't have to be equal to one. Test your answer, both by itself and with `LUsolve()`, and turn in the resulting Python code onto Gradescope (this is the one problem in this assignment where you have to turn in Python code). Your TAs can explain further about file formats and Gradescope in your weekly section. *Hint*: Loops can be run backward in Python, say from $n-1$ down to 0, by writing

```
for i in reversed(range(n)):
```