Assignment 2

Name:___________________; UID:___________________

1 Question 1

1) Given a matrix $A \in \mathbb{R}^{n\times n}$. At most how many additions (subtractions) and multiplications (divisions) are needed to order to perform the LU factorization for $A = LU$? Justify your answer.

2) Write code to realize the LU algorithm for factorizing a square matrix $A \in \mathbb{R}^{n\times n}$. Do an example by randomly generating a 100×100 matrix A (such as torch.randn) and LU factorize it.

2 Question 2

1) Given a lower triangular matrix L, write code to calculate its inverse. Analyze the computational complexity (the number of additions/multiplications needed)

2) If A is nonsingular, prove that the LU decomposition of A is unique.

3 Question 3

The lower triangular matrix $L \in \mathbb{R}^{n\times n}$, $n \geq 2$, is nonsingular, and the vector $b \in \mathbb{R}^n$ is such that $b_i = 0, i = 1, 2, ..., k$, with $1 \leq k \leq n$. The vector $y \in \mathbb{R}^n$ is the solution of $Ly = b$. Show, by partitioning L, that $y_j = 0, j = 1, 2, ..., k$. Hence give an alternative proof of Theorem 2.1(iv), that the inverse of a nonsingular lower triangular matrix is itself lower triangular.

4 Question 4

In practice, we do the pivoting technique in the following way: we find P_1 such that P_1 changes the first row with the r^{th} row where $|(A)_{r1}| = \max_{1\leq kleqn} |(A)_{k1}|$, then we find L_1 such that the first column of L_1P_1A contains all 0 except for the element in the first row. Then we repeat the process for $L_1P_1A(2:n, 2:n)$ (the matrix containing the second to the last columns and rows of L_1P_1A) by finding P_2, L_2...till we get an upper triangular matrix $L_nP_n...L_1P_1A$.

Write code to implement the following

- Using the above pivoting technique to LU factorize a matrix $A \in \mathbb{R}^n$. Try to do several examples by yourself. Here, LU factorization means to find P_i, L_i, and U (you don't need to calculate $L_n P_n ... L_1 P_1$)
- Using the above pivoting technique, after LU factorize a matrix A, solve the linear system $Ax = b$. Try to do several examples by yourself.

Analyze the computational cost. That is, the number of operations needed.

5 Question 5

Assume that $S, T \in \mathbb{R}^{n \times n}$ are two upper diagonal matrices. λ is a real number, I_n is the identity matrix such that the matrix $ST - \lambda I$ is nonsingular. Please give an algorithm that could solve $(ST - \lambda I)x = b$ with the number of operations being $O(n^2)$. Please justify your answer. That is, to analyze the number of operations needed.

6 Question 6

Using the LU factorization, give an upper bound on the computational cost for calculating the inverse of a $n \times n$ nonsingular matrix. (the total numbers of multiplications/additions needed?)