CS 530 - Fall 2019
Homework 2

Due: Tuesday, October 1 - to be submitted via Gradescope

Reading : Chapter 35, Sections 1, 2 and 3, pages 1106-1122.

1. LU Decomposition

Show the LUP decomposition of the 4×4 matrix M.

```
    3  2  1  9
M = 6  4  9  12
    9  0  6  3
    0  1  3  5
```

2. Uniqueness of decomposition

In class we mentioned that if a non-singular matrix M has an LU decomposition, then the decomposition is unique. That is the there is only one pair (L,U) with M = LU. You can find a short proof of this fact at: "http://www.cs.bu.edu/fac/homer/530f19/hw/unique-LU.pdf"

(i). Is this same result true for every singular M which has an LU decomposition? Briefly explain why or why not.

(ii). LUP decompositions are not unique. Give an example of a non-singular 3×3 matrix A for which there are two different LUP decompositions. (No proof needed here, just write A and the L, U, P and L', U' and P' which show this.)

3. Finding an inverse

```
Define T =    1 -1  0 0
             -1  4 -1 0
              0 -1  4 0
              0  0 -1 4
```

(i). Find an LU decomposition for T.

(ii). Use (i). to find the inverse of T.

-

4. Reducing general (non-square) matrix multiplication to matrix inversion.

Let I be an algorithm which takes as input a matrix M and outputs its inverse M^{-1}. In class we showed how you can multiply two $n \times n$ matrices A and B by inverting a $3n \times 3n$ matrix C.

(i). Show how to do the same thing for non-square matrices A and B where A is $n \times r$ and B is $r \times s$ for any positive natural numbers n, r, s. (Hint: Use padding of matrices with 0's, sort of similar to what you may have done for Strassen's algorithm.)

(ii). Give an example of how this would work when A is 4×2 and B is 2×3. Specifically, find the corresponding matrix C whose inverse would /give you AB.

5. A problem about random permutations

Consider the following randomized algorithm R which produces a permutation of {1,2,...,n} when run.

Algorithm R:

0. R starts with a vector V = (1,2,3,...,n) of the first n natural numbers specifying the identity permutation. You then "randomize" V by,
1. Independently at random pick n integers a1, a2, ..., an from the set 1,2,3,...,n. (This is done with replacement. That is, the integers you pick may be repeated in the list of ai's.)
2. For each element ai from i=1 to n, switch i in V with ai in V.
3. The result of the n switches in V is a permutation of the integers 1 through n which is the output of this randomized algorithm.

Call a permutation of 1,2,3,...,n random if the probablility that it is output by R is 1/n!.

Questions 1: Show that any one of the n! different permutations of 1,2,3,...,n could be output by some choice of a1, a2, ..., an in step 1 of algorithm R.

Question 2: True or false: An output of step 3 of algorithm R is a random permutation of 1,2,3,...,n? Briefly explain your answer.

Note: What you need to determine here is whether all of the n! permutations of V are equally likely to be output in step 3 of the algorithm.

Now change step 1 of algorithm R by not allowing repetitions of numbers in a1, a2, ..., an. That is we first choose a1 randomly from 1, 2,...n, but then choose a2 randomly from all the first n numbers except for a1, then choose a3 randomly from numbers 1,2,...n except for a1 and a2, etc.

Steps 2 and 3 of the algorithm remain the same.

Question 3: Answer the same question 2 for this slightly changed version of R, and explain your answer.