CS 530 - Fall 2018
Homework 2

Due: Thursday, October 4

Reading : Read the first 2 sections, (6 pages, up to page 162) of the posting on the course homepage on probabilistic algorithms.
The url is: http://cs-www.bu.edu/faculty/homer/530f18/rand-alg1.pdf

Problems:

1.

```
         2 2 1
Let M =  6 6 5
         0 1 4
```

i. Compute the determinant of M.

ii. Use the LU decomposition algorithm to try to determine the LU decomposition of M. If the algorithm does give the decomposition, write the L and U you found. If the algorithm does not work here, show where you get stuck.

iii. Does the matrix M have an LUP decomposition ? Briefly explain why or why not.

2. In this problem we want to show that computing the determinant of a matrix reduces to computing the LUP decomposition of that matrix.

(i). So you are give a matrix A and its decomposition into the matrices L,U and P with PA=LU as we've seen in class. (A may be or may not be non-singular.)
Use the decomposition to efficiently compute the determinant of matrix A. By efficiently I mean your method should be faster than using the definition of the determinant directly.

(ii). Now assume that T(n) = the number of steps the LUP algorithm takes to find the LUP decomposition of any non-singular n x n matrix A. Estimate the number of steps D(n) your algorithm uses when it starts with matrix A and it then computes the LUP decomposition, and finally uses the decomposition to to compute the determinant for matrix A in terms of n and the function T(n).

You should make some assumption about the function T(n) and its growth rate.
This was similarly done in Theorem 28.1 of our text for the function I(n). Take a look at Thm. 28.1 and use similar assumptions about T(n). For example, it is assumed in 28.1 that I(3n) = O(I(n)). State what assumptions you use to get the complexity of I(n) in terms of T(n) in this problem.

3. Solving systems of dependent equations.
Below is an A and b which give an example of 3 equations with 3 unknowns of the usual form Ax = b, but where the rank of your 3x3 matrix A is 2 (so A is singular).

```
      2  3  -2             5
A =   4  4  -2   and b =   3
     -2 -1   0             1
```

(i). Try to apply the LUP decomposition algorithm to A. Do you get an LU or an LUP decomposition for A ?
(ii). Do you think this same outcome happens every time you start with an A which is singular as above ? (Note that b does not play any role in your answer here.)

Specifically, state how many different solutions to these 3 equations there are when you fix b to be (5 3 1) as above ? explain your reasoning.
Would a different choice for b result in a different number of solutions to Ax=b?

4. Producing random permutations of V= $\{1, 2, ..., n\}$.
It should be possible for a a randomized algorithm R to produce every permutation of V, and have the property that each of the the n! permutations of V should be produced with the same 1/n! probability. So when you run R once, you get a random permutation of V.

Consider the following R.
Begin with a fixed permutation σ of the set V= $\{1, 2, ..., n\}$. (If you like just let σ be the identity function.)
You then "randomize" σ by:
1. Independently at random pick n integers $a_1, a_2, ..., a_n$ from the set V. (The integers you pick may be repeated in the list of a_i's.)
2. For each element i of V from 1 to n, swap $\sigma(i)$ with $\sigma(a_i)$ in V.
3. Output the σ that results.

(i). Is every permutation of {1,2,... n} generated by some run of this algorithm R? Why or why not?

(ii). Explain why the permutation that is output in step 3 is not a random permutation of V.