CS 630 - Fall 2020
Homework 3

Due: Monday, November 2

Reading : The rest of the reading on probabilistic algorithms found at
http://www.cs.bu.edu/fac/homer/630/rand-alg1.pdf
In particular read pages 7-9 of the reading.

Problems:

1. 1. You are given an algorithm A which can compute the product LU of any n x n lower triangular matrix L mulitplied by any n x n upper triangular matrix U in O($n^2(logn)$)steps.

i. Show how to use algorithm A to define an algorithm G which computes the product of any two n by n matrices M1 and M2.

State the big-O complexity of G and explain how you obtain that complexity for G. Is the big-O complexity of G bigger than, equal to, or smaller than that of A?

2. Do Exercise 7.2 on page 163 of the randomized algorithms reading.

3. This problem follow the first half of the proof of the Schwartz-Zippel lemma from the Probabilistic algorithms book. It does this for the specific polynomial f(x,y) give in part (i) below. Reading the S-Z Lemma on page 165 and the first half of the induction proof of the lemma on page 166 should help you in doing this problem, and then in understanding the proof of the S-Z Lemma.

(i). Consider the 2-variable function f(x,y) over the rational numbers Q defined by f(x,y) = $x^4y^2 - (1/3)x^3y^3 - 2x^3y^6 - 15x^2 + (5/8)y$.

Assume that as in the proof of the S-Z lemma for f(x,y) we have chosen a set S $\subseteq$ Q with $|S| = 8$. In fact let's assume S = { -1, 0 , 1 , 2, 3 4 , 5, 6} And we choose r_1 and r_2 uniformly at random from S.

Answer the following 3 questions:

What is the total degree d of f and which terms of f have this total degree ?
What is the highest power k of x in f? And which term(s) of f have x^k as a factor.

(Note: We know that if x appears in f at all (with any non-zero any coefficient), then k $>$ 0.)

(ii). Now factor out the variable x from f(x,y) to obtain f(x,y) $= x^0 f_0(y) + x^1 f_1(y) + ... + x^k f_k(y)$

.

In this case what are the k+1 functions $f_0(y), f_1(y), ...f_k(y)$? Write them out explicitly,

(iii). Finally, in this part the problem the goal is to show one of the two conclusions which enable the S-Z proof to work. Namely show that the prob($f_k(r_2) = 0$) is at most (d-k)/$|S|$.

Proof: To show this, note that $f_k(y)$ is a 1-variable polynomial which is not $\equiv 0$, by the definition of k, and the total degree of $f_k(y) \leq$ d-k. So the Prob($f_k(r_2) = 0$) $\leq$ d-k.

Explain and justify why this 2 line Proof above is correct in your own words.

4. Note: The graphs in this problem are standard graphs, not multigraphs. That is, for any two vertices there is at most 1 edge connecting them.

(i). Prove that for any integer k, there is a connected graph G with the property that the degree of any vertex in G is at least k times larger than the size of the min cut of G.
Note: You can do this by defining G_k giving its vertices and edges or by describing (or drawing) a sequence of of graphs G_k which makes it clear how the sequence goes on.

(ii). Give an example of a connected graph H with at least 3 edges whose opt mincut has exactly 1/2 of its vertices on each side of any optimal min-cut. What is the mincut in your graph H? Is the mincut in H unique ? Explain your answer.

Now consider the max-cut problem for the graph H. Find one max-cut in your graph and state the size of the maxcut.

(iii). Give an example of a connected graph I with $|V| \geq 3$ and with at least $|V|$ edges whose max cut size equals its min cut size. Write out the max cut of I and the min cut of I.

5. Let M be a an n by n matrix of 0's and 1's. M(i,j) is the entry in row i column j.

We call M rearrangeable if there is a sequence of row switches and column switches of M which result in all 1's along the diagonal of the matrix.

a. Give an example of a 3 x 3 M which is not rearrangeable but which has at least one 1 in every row and in every column.

b. Write an efficient (polynomial number of steps) algorithm to decide if a matrix M is rearrangeable.

One way to construct such an algorithm is to use M to define a bipartite graph G = (L,R,E) with n vertices in its L set and n vertices in its R set and E as its edge.
Now prove that M is rearrangeable iff its graph G has a perfect matching.

Note: If you use a different proof than the one I've suggested, explain why it is correct.

Then use this fact to conclude that there is a polynomial time algorithm as asked for in this problem. State what the big-0 complexity is of your algorithm and explain why this is its complexity.

You should explain

c. Show how your algorithm works and what answer it gives on the 4 by 4 matrix M given by

```
0 1 1 0

1 0 0 1

1 0 0 0

0 1 0 0
```