CS 530 - Fall 2018 Quiz 2

Instructions: Please answer any 3 of the following 4 questions. Each question counts equally. Write your answers on the Answersheet. If need be, you can also use 1 piece of paper for longer answers. Put an X over the number of the problem on the answersheet which you do not want graded.

1. (**Multiple choice - no partial credit.**) Each problem (i), (ii), and (iii) is worth 3 points. It is possible that in some questions there are several correct answers, or none. In that case give all of the answers that are correct.

   (i). Recall the randomized algorithm which can be used to test (probabilistically) if two polynomials f(x) and g(x) are identical. Assume in this case we are explicitly given the polynomial f(x) $= 2x^8 - 3x^5 + 4x^2 - 2x$.

   However now neither g(x) nor its degree is explicitly given. Instead we are given a "black box" program which we can use to find values of the polynomial g(x) by inputting a number t to the program which immediately tells us the value g(t).

   Assume that the random inputs to the possible answers below are chosen from a set S of 100 different numbers.

   Say whether each of A, B, and C is true or false.

   A.If we are told that the degree of g(x) is 6 and we evaluate the f and g on 8 random inputs from S and we find that they are equal on 4 of these inputs then we know with certainty that they are not equal.

   B. If we evaluate f(x) and g(x) on 8 random inputs from S and we find that they are equal on all 8 of these inputs then we know with certainty that they are equal.

   C. Now assume that f $\neq$ g(x) for some x. If we evaluate f(x) and g(x) on 4 different random inputs r's from S and we find that f(r) and g(r) are equal for all of them, and we then pick a fifth number r_5 at random from S, then prob($f(r_5) = g(r_5)$) $\leq .1$.

   (ii). Freivalds algorithm for matrix multiplication checking when the matrices are not all square is the same as before as long as the dimensions of the three matrices match up.

   Consider the algorithm when run on a 3 input matrices K, L and M when K is 3×2, L which is $2{\times}3$ and M which is $3{\times}\ 3$, as given below. Note that in this case a random binary column vector V of 3 bits is generated to test whether K$\times$L = M.

```
    5  2          L=    0 1 0          4  6  1
K= -1  0                3 2 0     M=   0  1  0
   -2  1                               3  0  0
```

   A. What is the output of Freivalds algorithm when the vector V with 3 rows is (0 1 1) ?

   B. In this case, is the output of the algorithm correct or incorrect ? Explain why.

   C. What is the error probability of Freivalds algorithm for this particular K, L and M ?

(iii). Say whether each of A, B, C is true or false.

Consider the contraction algorithm for the min-cut problem when run once on any graph G with n vertices.

A. For all G with at least 4 edges, the probability of finding a min cut in G is not equal to 1.

B. For all G with at least 9 vertices, the probability of finding a mincut in G is at least 1/25.

C. For all G, if G has at least 4 edges and has min-cut of size 2 then it also has a some cut of size 3.

2. **(Farms.)** Consider a class of mathematical objects called 'Farms', denoted by $\mathcal{F}$. Each farm F is associated with a unique (univariate) polynomial $P_F(x)$. In addition, for farms F_1 and F_2, $F_1 \neq F_2$ if and only if $P_{F_1}(x)$ and $P_{F_2}(x)$ are different polynomials. Given a farm F, the k-th power of F, for k a natural number greater than 0, is another farm denoted by F^k. Farms and their powers satisfy the following nice property:

$$P_{F^k}(x) = P_F(x^k)$$

Design a randomized algorithm that, given two polynomials $P_{F_1}(x)$ of degree d_1 and $P_{F_2}(x)$ of degree d_2 as inputs, checks whether $F_1 = F_2^k$. Your algorithm should have error probability less than 1/8. You can use the ideas from about polynomial identity testing from the class in your algorithm. You should briefly explain why the algorithm has the 1/8 error bound.

3. **(Biased coins.)** Assume you have a fair coin c, so prob(c=heads) = prob(c=tails)= 1/2.

(i). You want to use this coin c to simulate a biased coin b where prob (b=heads) = 3/8 and prob(b=tails) = 5/8. Describe how to do this using your coin c.

(ii). Do the same thing as in part (i), but for a biased coin b where prob (b=heads) = 3/10 and prob(b=tails) = 7/10.

4. **(Min-cut)** Consider the graph G below.

(i). (3 points)

What is the size of a min-cut of G ? How many min-cuts does G have ?
What is the size of the largest cut in G?

(ii). (6 points)

Give an example of a graph G_n with n vertices and the property that G_n has at least n(n-1)/2 different min-cuts.

Explain why your graph works.