Homework 2

Due: Tuesday, July 9, 4:00pm

CS 70: Discrete Mathematics and Probability Theory, Summer 2013

1. [9 points] For each of the following claims, state whether the claim is true or false, and give a short proof to justify your answer. (Note: By a stable marriage instance we mean an input for stable marriage, i.e., the sets of men and women and their preference lists.)

   1a. [3 points] In a stable marriage instance, if man M and woman W each put each other at the top of their respective preference lists, then M must be paired with W in every stable pairing.

   1b. [3 points] In a stable marriage instance with at least two men and two women, if man M and woman W each put each other at the bottom of their respective preference lists, then M cannot be paired with W in any stable pairing.

   1c. [3 points] For every $n \geq 2$, there exists a stable marriage instance with n men and n women that has an unstable pairing in which every unmatched man-woman pair is a rogue couple.

2. [12 points] This question concerns stable marriages and the propose-and-reject algorithm.

   2a. [5 points] Consider the following stable marriage instance.

Man	Women		
1	A	B	C
2	C	A	B
3	B	C	A

Woman	Men		
A	2	3	1
B	1	2	3
C	3	1	2

   List all the rogue couples for the following pairing: (1,C), (2,A), (3,B)

   2b. [7 points] Run the propose-and-reject algorithm (with men proposing) on the following instance. Use the same notation as on page 3 of Note 4 to illustrate the operation of your algorithm at each stage of the process. Show clearly the final stable pairing produced by the algorithm.

Man	Women			
1	B	A	C	D
2	A	C	B	D
3	C	B	A	D
4	B	A	C	D

Woman	Men			
A	3	4	1	2
B	2	3	4	1
C	4	1	2	3
D	1	2	3	4

3. [13 points] Suppose we are given n medical students and m hospitals. Each hospital h has some number q_h of slots, and we assume that the total number of students is larger than

the total number of slots (i.e., $\sum_{h=1}^{m} q_h < n$). Each student ranks the m hospitals in order of preference, and each hospital ranks the n students. The goal is to find an assignment of students to slots (one student per slot) that is stable in the sense that there is no rogue student-hospital pair. An unmatched student-hospital pair (s, h) is considered rogue (with respect to an assignment of students to slots) iff s would prefer h to her current situation (she is either unmatched or matched to a hospital she likes less than h) and h would prefer s over one of the students assigned to h. There are two main differences between this problem and the Stable Marriage Problem: (i) there are more students than slots, and (ii) each hospital generally has more than one slot.

3a. [6 points] Modify the propose-and-reject algorithm from class so that it finds a stable assignment of medical students to slots. (Hint: Let the students propose to hospitals, and let each hospital h maintain a waitlist of q_h provisionally accepted students.)

3b. [7 points] Give a succinct proof that your algorithm does indeed find a stable assignment. Your proof should involve the following version of the Improvement Lemma (see Lecture Note 4): For each hospital, after its waitlist becomes full, its least favorite student on the waitlist can only improve over time.

4. [10 points] Recall that the extended Euclidean algorithm allows you to compute $GCD(x, y)$ and express $GCD(x, y)$ as an integer linear combination of x and y (that is, $a \cdot x + b \cdot y$ for some integers a, b).

4a. [5 points] Use this algorithm to compute $GCD(4725, 273)$ and express $GCD(4725, 273) = a \cdot 4725 + b \cdot 273$ for some integers a, b. Show all the steps of the algorithm.

4b. [5 points] Prove that for all $x > 0$ and $y > 0$, $GCD(x, y)$ is the *smallest* positive number that can be written as an integer linear combination of x and y. (You should only need a couple sentences to prove this.)

5. [21 points] This problem will give you practice with modular arithmetic.

5a. [3 points] Evaluate $(3002 + 6002 \times 9002) \bmod 3$. Show your work. Do it the easy way! Order of operations is as usual.

5b. [3 points] Similarly to part 5a, evaluate $\left(1002^3 - 2468 \times 17 + 4\right) \bmod 5$.

5c. [3 points] The numbers $\{0, 1, 2, \ldots, 19\}$ are "representatives" of the congruence classes mod 20. For each of these classes, determine whether it has an inverse mod 20, and if so, state the inverse. You may use brute force (no need for an algorithm).

5d. [3 points] Evaluate $\frac{5-(19\times 3)}{7\times 9}$ in modulo 20 arithmetic. Show your work.

5e. [3 points] Use the extended Euclidean algorithm to find the inverse of 36, mod 55. Show all the steps of the algorithm.

5f. [3 points] Describe all the integer solutions to the equation $17x \equiv 4 \pmod{20}$. (This should be a single congruence class mod 20).

5g. [3 points] Solve the following pair of simultaneous equations mod 19, showing your work:

$$5x + 3y \equiv 0 \pmod{19}$$

$$y \equiv 4 + 12x \pmod{19}$$

6. [15 points] In this problem, you will discover how to solve individual linear equations in "mod m" arithmetic.

6a. [9 points] Consider the equation $ax \equiv b \pmod{m}$, where x is the unknown and a, b, m are given (with $m > 0$). Prove that this equation has either no solutions mod m, or exactly d solutions mod m, where $d = GCD(a, m)$; also, describe when each of these two cases holds. (Hint: Consider the (non-modular) integer equation $ax - km = b$ (for some integer k), and consider dividing by d.)

6b. [6 points] Using your answer from the previous part, describe all the solutions mod 63 of each of the following three equations:

(i) $4x + 28 \equiv 2 \pmod{63}$

(ii) $7x + 50 \equiv 35 \pmod{63}$

(iii) $7x + 50 \equiv 36 \pmod{63}$

7. [12 points] Fermat's Little Theorem states that, if p is prime, then for every $a \in \{1, 2, \ldots, p-1\}$, we have $a^{p-1} \equiv 1 \pmod{p}$. This theorem is a key ingredient in the proof of correctness of the RSA cryptosystem, and is useful for many other things (and it often shows up on CS70 exams).

7a. [7 points] Prove Fermat's Little Theorem. (Hint: Show that the set of $p - 1$ numbers $\{a \cdot 1,\ a \cdot 2,\ \ldots,\ a \cdot (p-1)\}$ are all distinct and non-zero mod p. Then multiply them together.)

7b. [5 points] Prove the following generalization of Fermat's Little Theorem: For every positive integer n (not necessarily prime), let S_n be the set of integers $a \in \{1, 2, \ldots, n-1\}$ such that $GCD(a, n) = 1$. Then for every $a \in S_n$, we have $a^{|S_n|} \equiv 1 \pmod{n}$. (Here $|S_n|$ denotes the number of elements in S_n.)

8. [8 points] A *pseudorandom number generator* is a way of generating a large quantity of random-looking numbers, if all we have is a little bit of randomness (known as the *seed*). One simple scheme is the *linear congruential generator*, where we pick some modulus m, some constants a, b, and a seed x_0, and then generate the sequence of outputs $x_0, x_1, x_2, x_3, \ldots$ according to the following equation:

$$x_{t+1} = (a \cdot x_t + b) \bmod m$$

(Notice that $0 \le x_t < m$ holds for every t.) You've discovered that a popular web site uses a linear congruential generator to generate poker hands for its players. For instance, it uses

x_0 to pseudo-randomly pick the first card to go into your hand, x_1 to pseudo-randomly pick the second card to go into your hand, and so on. For extra security, the poker site has kept the parameters a and b secret, but you do know that the modulus is $m = 2^{31} - 1$ (which is prime). Suppose that you can determine the values x_0, x_1, x_2, x_3, and x_4 from the information available to you, and that the values $x_5, \ldots, x_9$ will be used to pseudo-randomly pick the cards for the next person's hand. Describe how to efficiently predict the values $x_5, \ldots, x_9$, given the values known to you.