CS 70 Discrete Mathematics and Probability Theory
Summer 2015 Chung-Wei Lin HW 2

Due Monday July 6 at Noon

1. **Stable Teaching Assistant** (20 points, 5 points for each part)

   You have been asked to assign TAs for the fall semester. Each class has its own method for ranking candidates, and each candidate has their own preferences. An assignment is **unstable** if a class and a candidate prefer each other to their current assignments. Otherwise, it is **stable**.

   Candidate information:

Candidate	CS70 Grade	CS61A Grade	CS61B Grade	Teaching Experience	Overall GPA	Preferences
A	A+	A	A	Yes	3.80	CS70 > CS61A > CS61B
B	A	A	A	No	3.70	CS70 > CS61B > CS61A
C	A	A+	A-	Yes	3.60	CS70 > CS61A > CS61B

   Ranking method:

   - CS70: Rank by CS70 grade. Break ties using teaching experience, then overall GPA.
   - CS61A: Rank by teaching experience. Break ties using CS61A grade, then overall GPA.
   - CS61B: Rank by CS61B grade. Break ties using overall GPA, then teaching experience.

   (a) Find a stable assignment.

   (b) Can you find another, or is there only one stable assignment (if there is only one, why)?

   CS70 is overenrolled and needs two TAs. There is another candidate.

Candidate	CS70 Grade	CS61A Grade	CS61B Grade	Teaching Experience	Overall GPA	Preference
D	A+	A	A+	No	3.90	CS61A > CS61B > CS70

   (c) Find a stable assignment.

   (d) Prove your assignment in Part (c) is stable.

2. **Queer Stable Marriage** (5 points)

   Because people in the US can now marry any gender, we need another way to give them a stable pairing, which is essentially the stable roommates problem. An online dating app Dating70 wants to match up Jerry, Eric, Sam, and Lucy. They give the app the following preferences for their matches:

Jerry	Eric $>$ Lucy $>$ Sam
Eric	Sam $>$ Jerry $>$ Lucy
Sam	Jerry $>$ Eric $>$ Lucy
Lucy	Jerry $>$ Eric $>$ Sam

An algorithm consists of each person x, one by one, proposing to the first person y on their list and executing as follows:

- When y is proposed by x, y crosses off everyone below x on y's list.
- If y holds 2 proposals, y rejects the person y prefers least (crosses off the person on y's list).
- When x is rejected by y, x crosses off y on x's list and proposes to the next person immediately.

This continues until everyone holds exactly one proposal. We start with the following proposals and produce the following table:

Jerry	$\rightarrow$	Eric, Eric crosses off Lucy
Eric	$\rightarrow$	Sam, Sam crosses off Lucy
Sam	$\rightarrow$	Jerry
Lucy	$\rightarrow$	Jerry, Jerry rejects/crosses off Sam, Sam crosses off Jerry
Sam	$\rightarrow$	Eric, Eric rejects/crosses off Jerry, Jerry crosses off Eric
Jerry	$\rightarrow$	Lucy, Lucy crosses off Eric and Sam

Jerry	~~Eric~~ $>$ Lucy $>$ ~~Sam~~
Eric	Sam $>$ ~~Jerry~~ $>$ ~~Lucy~~
Sam	~~Jerry~~ $>$ Eric $>$ ~~Lucy~~
Lucy	Jerry $>$ ~~Eric~~ $>$ ~~Sam~~

Since each person only has a list size of one, the algorithm terminates with the pairing: {(Jerry, Lucy), (Eric, Sam)}.

Now Megan Fox and Angelina Jolie decide to join the group. Try the algorithm on the following table to find a pairing:

Jerry	Eric	$>$	Megan	$>$	Lucy	$>$	Sam	$>$	Angelina
Eric	Megan	$>$	Sam	$>$	Jerry	$>$	Angelina	$>$	Lucy
Sam	Jerry	$>$	Megan	$>$	Eric	$>$	Lucy	$>$	Angelina
Lucy	Angelina	$>$	Jerry	$>$	Eric	$>$	Sam	$>$	Megan
Megan	Jerry	$>$	Angelina	$>$	Eric	$>$	Lucy	$>$	Sam
Angelina	Sam	$>$	Lucy	$>$	Megan	$>$	Jerry	$>$	Eric

Note: The output of this example will be a stable pairing. However, for any instance, if it has a stable pairing, the algorithm cannot guarantee to find the stable pairing. In fact, the algorithm described above is only the Phase 1 of the Irving Algorithm. With the Phase 2, the Irving Algorithm can always find a stable pairing, if the given instance has one. For more information, please check https://en.wikipedia.org/wiki/Stable_roommates_problem

3. **System Equations with Modular Arithmetic** (20 points, 2 points for each part)

   (a) Solve the following system equations:

   $$\begin{cases} x+y=9 & \ldots \text{Equation A} \\ x-y=5 & \ldots \text{Equation B} \end{cases}$$

   For the following parts (b)–(h), we will consider mod 5, and only numbers in $\{0,1,2,3,4\}$ and notations $\{x, y, +, =\}$ can be used in the answers.

   (b) Rewrite Equation A. (Assume the answer as Equation C.)

   (c) Rewrite Equation B. (Assume the answer as Equation D.)

   (d) Write down $(4 \times (\text{Equation C}) - (\text{Equation D}))$.

   (e) Solve x from Part (d).

   (f) Write down $((\text{Equation C}) - (\text{Equation D}))$.

   (g) Solve y from Part (f).

   (h) Rewrite the answer from Part (a).

   (i) Compare the results from Parts (e), (g), and (h). What do you find?

   (j) Is $(7,7)$ a solution to the original system equations? Is it a solution if we consider mod 5?

4. **Mod Never Bothered Me Anyway** (15 points, 5 points for each part)

   (a) Run the Extended Euclid's Algorithm: `egcd`(31,21).

   (b) In Arendelle (the name of a fictional country), there are only two types of coins: one worth 31 cents and the other one worth 21 cents. Elsa used coins to buy a gift for Anna and paid 1,010 cents exactly. How many 31-cent coins and 21-cent coins did Elsa use?

   (c) Prove: there is only one solution.

5. **The Bad Bartender** (15 points, 5/10 points for each part)

   You're well-known for being a decent bartender at school, so you get hired at an alumni event. The task is deceptively simple — just pour the jungle juice!

   The party has started, but you only have a 3-ounce tumbler, a 5-ounce tumbler, and a big bowl of juice. The alumni are very picky — they want an exact amount of liquid in their cups.

   You are allowed to pour from one container to another, stopping only when either the pouring container is empty or the receiving container is full. Each of the alumni holds their own cup. You cannot use the alumnus's cup when preparing the exact amount of juice.

   (a) Your first alumnus wants 4 ounces of jungle juice. Show how to pour exactly the right amount.

   (b) Given an a-ounce tumbler (a is an integer) and a p-ounce tumbler (p is a prime) where $a \neq p \pmod{p}$, design an algorithm to get all possible integer $\{0,1,\ldots,p\}$ ounces of juice. Reasonably explain why your algorithm works.

6. **Generalization of Fermat's Little Theorem** (15 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 integer $a \in \{1, 2, \ldots, n\}$ and $GCD(a, n) = 1$. 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.)

7. **RSA** (30 points, 5/5/5/5/10 points for each part)

   Consider an RSA scheme modulus $N = pq$, where p and q are prime numbers larger than 3. In this setting, Alice wants to send a message x to Bob with public key (N, e).

   (a) Find a condition on p and q such that $e = 3$ is a valid exponent.

   (b) Now suppose that $p = 37$, $q = 13$, and $e = 17$. Find the secret key d used in this scheme.

   (c) Following Part (b), Bob receives the encrypted message $y = 91$. What was the original message Alice sent (before encrypting it)?

   (d) Now suppose that $p = 7$, $q = 3$, and $e = 25$. Alice wants to send a message $x = 8$ to Bob. What is the encrypted message she sends using the public key?

   (e) Prove: When $e \equiv 1 \pmod{p-1}$ and $e \equiv 1 \pmod{q-1}$, $x^e \equiv x \pmod{pq}$.