CS 70 SPRING 2007 — DISCUSSION #3

ALEX FABRIKANT

1. Administrivia

(1) Course Information
- Next homework is **due February 6th** at 2:30pm in 283 Soda Hall
- Please write down your section's time on the homework.
- Webpage for this section: `http://www.cs.berkeley.edu/~alexf/cs70/`

2. The Stable Marriage Problem

Recall from class the Stable Marriage Problem, and the associated propose and reject (a.k.a. the Traditional Marriage) algorithm. The following facts can be proven about the correctness of this algorithm:

Facts 1. *For the case when men propose and women accept/reject:*

(i) No man can be rejected by all women.
(ii) The sequence of proposals made by each man is non-increasing in his preference list.
(iii) The sequence of men that a woman holds on a string is non-decreasing in the woman's preference list.
(iv) The algorithm terminates with a stable matching.
(v) The propose-reject algorithm terminates in at most n^2 days.
(vi) The propose-reject algorithm always produces a male-optimal stable matching.
(vii) A male-optimal stable matching is a female-pessimal stable matching.

Exercise 2 (Warm-up)**.** You start out with the usual stable matching setup – n men, n women, and their preferences. Suppose you first run the traditional propose-and-reject algorithm, come up with some stable matching. Then, you run a "Sadie Hawkins" version of the algorithm, which is the exact same algorithm, but with women taking the role of men (proposing) and men taking the role of women (accepting/rejecting). The "Sadie Hawkins" version gives you the *exact same stable matching* as the traditional algorithm. What can you conclude?

Exercise 3. Try to recall the proof of each of these facts.

Exercise 4. To bring the stable matching problem a bit closer to home, consider a different problem which we'll call (oh, for no particular reason) the "CS Stable Matching Problem". The setup is the same, except that you have N women and M men, with $M > N$, so that some men will not be matched to women. We assume that not being matched at all is the least preferable choice for everyone.

We'll try to solve this by adding an extra $M - N$ "virtual women", $V_1, V_2, \ldots, V_{M-N}$ to the set of women. Getting matched to these will correspond to not getting matched to a woman, so the V_i's will rank behind all the "actual" women on all the men's preferences. Since V_i are "empty spots", let's set all their preferences to, say, the men in alphabetical order.

(1) Does running the traditional stable matching algorithm on this "virtual" problem produce a stable "virtual" matching? Why?
(2) Does the stable "virtual" matching have to correspond to a "real" stable matching in the original "CS stable matching" problem? Why?

Date: February 2, 2007.

The author gratefully acknowledges the TA's of CS70 Past for the use of their previous notes: Amir Kamil, Chris Crutchfield, David Garmire, Lorenzo Orecchia, and Ben Rubinstein. Their notes form the basis for this handout.

Man	highest→lowest			
1	B	A	V_1	V_2
2	A	B	V_1	V_2
3	A	B	V_2	V_1
4	B	A	V_1	V_2

TABLE 1. Men's preference list

Woman	highest→lowest			
A	3	1	2	4
B	2	4	1	3
V_1	1	2	3	4
V_2	1	2	3	4

TABLE 2. Women's preference list

(3) Running the stable matching algorithm with the preferences defined above produces a chain of proposal and rejection events with each man proposing to at most one woman per day and each woman saying "maybe" to at most one man per day.

Use an induction on "days" (steps of the stable matching algorithm) to show that, if you change the ordering of the virtual women's preferences in some way, this approach still produces the same "real" stable matching (i.e. the same matches are produced, the same set of men remain unmatched, and the only differences concern which of the "virtual women" is "matched" to which of the unmatched men).

3. Modular Arithmetic

3.1. **Notation.** In the last lecture, you saw this notation for "a has the same remainder as b when divided by n":

$$a \equiv b \mod n$$

Often (and, hopefully, in subsequent lectures), this is written as follows, for more clarity:

$$a \equiv b \pmod{n}$$

This clarifies that the $\equiv$ symbol represents that a and b are "the same" in the "modulo n" notion of sameness. These *congruences*, much like equalities in normal arithmetic, form a so-called *equivalence relation*. That is, they satisfy the following three properties:

(1) Reflexive: $a \equiv a \pmod{n}$
(2) Symmetric: $a \equiv b \pmod{n} \implies b \equiv a \pmod{n}$
(3) Transitive: $a \equiv b \pmod{n} \wedge b \equiv c \pmod{n} \implies a \equiv c \pmod{n}$

From the definition, we have:

$$a \equiv b \pmod{n} \iff n | (a - b)$$

Why is this true? Recall that if $a \equiv b \mod n$, then this means that $a = b + nk$ for some $k \in \mathbb{Z}$. Then clearly we have $nk = a - b$, from which it follows that n must divide $a - b$.

Equivalently, if you consider the C-like % (remainder) operator, then

$$a \equiv b \pmod{n} \iff a\%n = b\%n.$$

Mathematicians will also use the " mod " express this *operation on integers*:

$$a \equiv b \pmod{n} \iff a \mod n = b \mod n.$$

Hence, we can write: $4 \mod 3 = 1$, $15 \mod 7 = 1$, $0 \mod 4 = 0$, $7 \mod 2 + 20 \mod 3 = 3$, etc.
The difference between the two uses of mod is:

> $a \mod n$ is an operation that takes two integers and produces another integer, while $a \equiv b \pmod{n}$ is a statement about elements of the "modulo n" world ($\mathbb{Z}_n$) represented by a and b

3.2. **Operations in modular arithmetic.** Addition and multiplication in $\mathbb{Z}_n$ work the same as they do in $\mathbb{Z}$. The following rules hold:

(1) $a \equiv b \pmod{n} \implies a + c \equiv b + c \pmod{n}$
(2) $a \equiv b \pmod{n} \implies a - c \equiv b - c \pmod{n}$
(3) $a \equiv b \pmod{n} \implies a \cdot c \equiv b \cdot c \pmod{n}$

But what about division?

Let's consider what it means when you want to solve for x in the equation

$$2x \equiv 6 \pmod{8}$$

Well, clearly $x \equiv 3 \pmod{8}$ is a solution. But $x \equiv 7 \pmod{8}$ is also a solution! So, in a sense, division by 2 in $\mathbb{Z}_8$ is not well-defined (how can x be congruent to both 3 and 7?)

However, if we try and solve for x in the problem

$$3x \equiv 6 \pmod{8},$$

the only solution is $x \equiv 2 \pmod{8}$ (try it out!) Can you guess when you're allowed to divide and when you aren't?

Now let's move on to exercises[1]:

Exercise 5. We saw in class that the following statement can be proven by induction:

$$\forall a, b, n \in \mathbb{N}, a \equiv b \pmod{m} \implies a^n \equiv b^n \pmod{m}$$

Complete this proof.

Exercise 6. Show that if a is an odd integer, then $a^2 \equiv 1 \pmod{8}$.

Exercise 7. Show that if $n \equiv 3 \pmod{4}$, then n cannot be the sum of the squares of two integers.

4. Challenge problem

Exercise 8. Suppose you're actually a med student, and you're about to enter the residency matching. This will determine the next several years of your life and perhaps significantly alter your career path. In short, you really care about this.

(1) Suppose you're not only a med student, but also an omniscient (albeit not omnipotent) deity who knows the exact preference lists of all your fellow med students and all the hospitals. Of course, you also have your own preference list. May it be the case that by *lying about your preferences* to the matching algorithm, you would get matched with a better hospital than you would by being truthful? Assume the residency matching is student-optimal (hospital-pessimal).
(2) What if this was 30 years ago when it was hospital-optimal and student-pessimal?
(3) What if you're not a deity and have only partial information about your colleagues' and the hospitals' information? I leave it open-ended for you to define exactly how to define "partial information".

Disclaimer: After a very cursory web search, I do not have concrete evidence that these problems have thus far been solved. While the first two parts may well have a tractable, known solution, it's very likely that there are some interpretations of the last part that are open research problems. If you make any progress on any of these, please do let me know. As you can surely guess, any new results on this problem would be very interesting, and not only to med students. Also, especially if you attempt the third part, I encourage you to search for existing work beforehand. Google and Google Scholar are good starting points.

[1]Exercises are from Rosen's *Elementary Number Theory* and Dasgupta, Papadimitriou and Vazirani's *Algorithms*