University of Washington
Department of Computer Science and Engineering
CSE 421, Autumn 2019

September 24, 2019

Homework 1, Due Wednesday, October 2, 1:29 pm, 2019

Turnin instructions: Electronic submission on canvas using the CSE 421 canvas site. Each numbered problem is to be turned in as a separate PDF, with subparts submitted together. E.g., Problems 2a and 2b should be turned in together.

Problem 1 (15 points):

Let $I = (M, W)$ be an instance of the stable matching problem. Suppose that the preference lists of all $m \in M$ are identical, so without loss of generality, m_i has the preference list $[w_1, w_2, \ldots, w_n]$. Prove that there is a unique solution to this instance.

Problem 2 (15 points):

Prove that the stable matching problem may have an exponential number of solutions. To be specific, show that for every n, there is an instance of stable matching on sets M and W with $|M| = |W| = n$ where there are at least c^n stable matchings, for some $c > 1$. (There is a solution with $c = \sqrt{2}$, but you can use a different constant.)

Programming Problem 3a (15 points):

Implement the stable matching algorithm. Write an input generator which creates completely random preference lists, so that each M has a random permutation of the W's for preference, and vice-versa. The goodness of a match for an individual can be measured by the position in the preference list of the match. The overall goodness for the M's would be the sum over each m, of his rank for the matching w. Similarly, the goodness for the W's can be defined.
You are free to write in any programming language you like. The quality of your algorithm may be graded (but you can use the one in the book), but the actual quality of the code will not be graded. The expectation is that you write the algorithmic code yourself - but you can use other code or libraries for supporting operations. You may use a library to generate random permutations (although this can be done as a four-line algorithm.) Submit your code as a PDF.
Make sure that you test your algorithm on small instance sizes, where you are able to check results by hand.

Programming Problem 3b (15 points) :

As the size of the problem increases - how does the goodness change for M and W? (It is probably easiest to normalize by dividing the goodness by n, the number of pairs.) Submit a write up about how the goodness varies with the input size based on your experiments. Can you determine the asymptotic growth rate? Is the result better for the M's or W's? You will probably need to run your algorithm on inputs with n at least 1,000 to get interesting results.

Problem 4 (20 points):

Gale and Shapley published their paper on the stable marriage problem in 1962; but a version of their algorithm had already been in use for ten years by the National Resident Matching Program, for the problem of assigning medical residents to hospitals.

Basically, the situation was the following: There were m hospitals, each with a certain number of available positions for hiring residents. There were n medical students graduating in a given year, each interested in joining one of the hospitals. Each hospital had a ranking of the students in order of preference, and each student had a ranking of the hospitals in order of preference. We will assume that there were more students graduating than there were slots available in the m hospitals.

The interest, naturally, was in finding a way of assigning each student to at most one hospital, in such a way that all available positions in all hospitals were filled. (Since we are assuming a surplus of students, there would be some students who do not get assigned to any hospital.) We say that an assignment of students to hospitals is stable if neither of the following situations arises:

- **First type of instability:** There are students s and s', and a hospital h, so that s is assigned to h and s' is assigned to no hospital, and h prefers s' to s.
- **Second type of instability:** There are students s and s', and hospitals h and h', so that:
  - s is assigned to h, and
  - s' is assigned to h', and
  - h prefers s' to s, and s' prefers h to h'.

So we basically have the stable marriage problem from class, except that (i) hospitals generally want more than one resident, and (ii) there is a surplus of medical students. Show that there is always a stable assignment of students to hospitals, and give an polynomial time algorithm to find one.

Problem 5 (20 points):

(From text, page 28, exercise 8.) For this problem, we explore the issue of *truthfulness* in the Gale-Shapley algorithm for Stable Matching. Can a participant improve its outcome by lying about its preferences. Consider $w \in W$. Suppose w prefers m to m', but m and m' are low on w's preference list. Is it possible that by switching the order of m and m' on w's preference list, w achieves a better outcome, e.g., is matched with an m'' higher on the preference list than the one if the actual order was used.

Resolve this question in one of two ways:

(a) Give a proof that, for any set of preference lists, switching the order of a pair on the list cannot improve a w's partner in the Gale-Shapley algorithm; or

(b) Give an example of a set of preference lists for which there is a switch that would improve the partner of a w who switched preferences.