University of Washington September 27, 2016
Department of Computer Science and Engineering
CSE 421, Autumn 2016

Homework 1, Due Wednesday October 5, 2016

Problem 1 (10 points):
Page 22, Exercise 1.

Problem 2 (10 points):
Page 22, Exercise 2.

Problem 3 (10 points):
Page 27, Exercise 8.

Programming Problem4 (30 points) :
(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.

(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.

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 will write the algorithmic code yourself - but you can use other code or libraries for supporting operations.

Make sure that you test your algorithm on small instance sizes, where you are able to check results by hand.