CS2800 Design and Analysis of Algorithms
Problem Set #1 January 23, 2023

1. Consider the following statement about the stable matching problem.

   > In every instance of the stable matching problem, there is a stable matching containing a pair (m, w) such that m is ranked first on the preference list of w and w is ranked first on the preference list of m.

   Decide if the statement is true or false. If it is true, give a brief explanation of why it is true. If it is false, then give a counter-example.

2. Give an instance of the stable matching problem and a stable matching for that instance that cannot be obtained by running the Gale-Shapley algorithm as we described in class.

3. Consider a generalization of the stable matching problem, where some students do not rank all the courses, and some courses do not rank all the students. Furthermore, a student is matched to a course only if each appears in the other's preference list. In this case, we have the following unstable situations apart from the one we saw in class.

   - The student s_i is matched to c_j, and there is an unmatched course c_k such that s_i prefers c_k to c_j.
   - The course c_j is matched to s_i, and there is an unmatched student s_k such that c_j prefers s_k to s_i.
   - The student s_i and the course c_j are not matched to anyone in the matching, yet they appear in each others preference lists.

   We will say that a matching is stable if none of the above cases arise. Note that in this setting, the stable matching may leave some students and courses unmatched even though they have a preference list. Describe how you will compute a stable matching by reducing the problem to an instance of the problem we saw in class and use the Gale-Shapley algorithm.

   Hint: Try to reduce this to an instance of the stable marriage problem seen in class, and apply the Gale-Shapley algorithm.

4. Let us try to understand if the Gale-Shapley algorithm is robust against malicious parties misrepresenting their preferences. Consider the scenario where a student knows the preference order of all the courses beforehand. Would it be favorable for the student to lie about his/her actual preference order to get a better match?

   In particular consider the following scenario. Suppose there is a student S who knows the preference order of all the other students and courses. Also, the student knows that the Gale-Shapley algorithm will be run with the students applying to the courses. Suppose the student realises that (s)he will be allotted a course C if the Gale-Shapley algorithm is run. Is it possible

for the student to give a different preference order than her/his actual one so that the Gale-Shapley algorithm ends up allotting a C' that is higher in her actual preference list? Either show that this is impossible, or construct a scenario where this happens.

What about if the Gale-Shapley algorithm is run with the students applying, and a course (that knows the preference orders of everyone else) decides to change its preferences to get a better student. Is this possible? Either prove that this is impossible, or construct a scenario where this happens.