CS3230 Problem Set 1

Due: Monday, 28th Aug 2023, 6 pm SGT.

Please upload a PDF file containing your solution (hand-written & scanned, or typed) by 28th Aug, 6 pm on Canvas. You may discuss the problems with your classmates, though you should write up your solutions on your own. Please note the names of your collaborators in your submission. You may want to refer to the plagiarism policy on Canvas.

Problems for Submission

1. Suppose that in an instance of the Stable Matching problem where a man m ranks a woman w as his first choice, and the woman w also ranks the man m as her first choice. Prove that in any stable matching, m and w will be matched.

2. In this problem, you will investigate what happens when participants cheat when reporting preferences to the Gale-Shapley algorithm.

   Consider the example discussed in lecture, where the men are Ashish, Bao, and Charlie, and the women are Xinyu, Yashoda, and Zuzu. Recall that Gale-Shapley results in the men-optimal stable matching: (Ashish, Xinyu), (Bao, Yashoda), (Charlie, Zuzu). Show that Xinyu can misreport her preferences so as to force Gale-Shapley to output the women-optimal stable matching: (Ashish, Yashoda), (Bao, Xinyu), (Charlie, Zuzu).

3. Read this article about how bias and unfairness can arise in algorithmic decisions. Write a paragraph summarizing the problem the article discusses.

   Optional: If you are interested in following up, try the assignment on "Criminal Justice" described here.

Additional Problems

1. Suppose that we are matching students to tutorials. There are n students and m tutorials. Each student is matched to at most one tutorial. Each tutorial has its own size limit; the i'th tutorial can accept at most s_i students for an integer $s_i \geq 0$.

   Each student has a ranking of the tutorials in order of preference, and each tutorial has a ranking of the students in order of preference.

   A matching M between students and tutorials is said to be *unstable* if there is a student x and a tutorial y such that both of the following hold:

   (i) Either x is unmatched, or x prefers y to the current assignment for x.

   (ii) Either y is not full, or y prefers x to one of the students currently assigned to y.

   Generalize the Gale-Shapley algorithm to find a stable matching in this situation.

2. In the setting of the previous problem, suppose that some student-tutorial assignments are absolutely forbidden due to schedule conflicts. Here, a matching M between students and tutorials is *unstable* if there is a student x and a tutorial y such that all three of the following hold:

   (i) Assigning x to y is not forbidden.

   (ii) Either x is unmatched, or x prefers y to the current assignment for x.

   (iii) Either y is not full, or y prefers x to one of the students currently assigned to y.

   Extend Gale-Shapley to this situation.

3. [**Problem 6, Chapter 1 in Kleinberg-Tardos**] Peripatetic Shipping Lines, Inc., is a shipping company that owns n ships and provides service to n ports. Each of its ships has a *schedule* that says for each day of the month which of the ports it's visiting, or whether it's out at sea. (You can assume the "month" here has m days for some $m > n$.) Each ship visits each port for exactly one day during the month. For safety reasons, PSL Inc. has the following strict requirement:

   (*) No two ships can be in the same port on the same day.

   The company wants to perform maintenance on all the ships this month, via the following scheme. They want to *truncate* each ship's schedule: for each ship S_i, there will be some day when it arrives in its scheduled port and simply remains there for the rest of the month (for maintenance). This means that S_i will not visit the remaining ports on its schedule (if any) that month, but this is okay. So the truncation of S_i's schedule will simply comsist of its original schedule up to a sertain specified day on which it is in port P; the remainder of the truncated schedule simply has it remain in port P.

   Now, the company question to you is the following: Given the schedule for each ship, find a truncation of each so that condition (*) continues to hold: no two ships are ever in the same port on the same day.

As an example, suppose there are two ships and two ports ($n = 2$), and the "month" has four days. Suppose the first ship's schedule is: *port* P_1*; at sea; port* P_2*; at sea*, and the second ship's schedule is *at sea; port* P_1*; at sea; port* P_2. Then, one can choose truncations so that the first ships stays at port P_2 starting on day 3 and the second ship stays at port P_1 starting on day 2.

Show that such a set of truncations can always be found, and give an algorithm to find them.

Hint: Convert the problem into an instance of stable matching between ships and ports.