CSCI 570 - Fall 2021 - HW 1

Due: *Sep 3rd*

1. Reading Assignment: Kleinberg and Tardos, Chapter 1.

2. Solve Kleinberg and Tardos, Chapter 1, Exercise 1.

   False. Suppose we have two men m, m_0 and two women w, w_0. Let m rank w first; w rank m_0 first; m_0 rank w_0 first; and w_0 rank m first. We see that such a pair as described by the claim does not exist.

   Rubric (5pt):

   - 2 pt: Correct T/F claim
   - 3 pt: Provides a correct counterexample as explanation

3. Solve Kleinberg and Tardos, Chapter 1, Exercise 2.

   True. Suppose S is a stable matching where m and w are not paired with each other. Suppose that contains the pairs (m, w_0) and (m_0, w) instead, for some $m_0 \neq m, w_0 \neq w$. Clearly, the pairing (m, w) is preferred by both m and w over their current pairings, contradicting the stability of S. Thus, by contradiction every stable matching must contain (m, w).

   Rubric (5pt):

   - 2 pt: Correct T/F claim
   - 3 pt: Provides a correct explanation

4. State True/False: An instance of the stable marriage problem has a unique stable matching if and only if the version of the Gale-Shapely algorithm where the male proposes and the version where the female proposes both yield the exact same matching.

   True.
   Proving the given statement requires proving the claims in two directions ('if' and 'only if').
   Claim: *"If there is a unique stable matching then the version of the Gale-Shapely algorithm where the male proposes and the version where the female proposes both yield the exact same matching."*

   The proof of the above claim is clear by virtue of the correctness of Gale-Shapely algorithm. That is, the version of the Gale-Shapely algorithm where the male proposes and the version where the female proposes are both correct and hence will both yield a stable matching. However since there is a unique stable matching, both versions of the algorithms should yield the same matching.

Next, we prove the converse: *"If the version of the Gale-Shapely algorithm where the male proposes and the version where the female proposes both yield the exact same matching then there is a unique stable matching."*

The proof of the converse is perhaps more interesting. For the definition of best valid partner, worst valid partner etc., see page 10-11 in the textbook.

Let S denote the stable matching obtained from the version where men propose and let S_0 be the stable matching obtained from the version where women propose.

From (page 11, statement 1.8 in the text), in S, every woman is paired with her worst valid partner. Applying (page 10, statement 1.7) by symmetry to the version of Gale-Shapely where women propose, it follows that in S_0, every woman is paired with her best valid partner. Since S and S_0 are the same matching, it follows that for every woman, the best valid partner and the worst valid partner are the same. This implies that every woman has a unique valid partner which implies that there is a unique stable matching.

Rubric (10pt):

- 4pts: Correct ' $\implies$ ' proof
  - (a) 1pt: Correctly state forward claim
  - (b) 3pt: Correctly justify forward claim
- 6pts: Correct ' $\impliedby$ ' proof
  - (a) 1pt: Correctly state backwards claim
  - (b) 5pts: Correctly justify backwards claim

5. A stable roommate problem with 4 students a, b, c, d is defined as follows. Each student ranks the other three in strict order of preference. A matching is defined as the separation of the students into two disjoint pairs. A matching is stable if no two separated students prefer each other to their current roommates. Does a stable matching always exist? If yes, give a proof. Otherwise give an example roommate preference where no stable matching exists.

   A stable matching need not exist. Consider the following list of preferences. Let a, b, c all have d last on their lists. Say a prefers b over c, b prefers c over a, and c prefers a over b. In every matching, one of a, b, c should be paired with d and the other two with each other. **Without loss of generality**, suppose a gets paired with d (thus, b, c with each other). Now, a prefers c over his current partner d and c also prefer to be with each other. Thus every matching is unstable no stable matching exists in this case.

Rubric (6pt):

- 1pts: Correct answer (i.e. matching need not exist)
- 5pts: Correct counter-example as explanation
  (a) 2pt: Describe the preference lists
  (b) 3pts: Explain why no stable matching exists

6. Solve Kleinberg and Tardos, Chapter 1, Exercise 3.

   We will give an example (with $n = 2$) of a set of TV shows/associated ratings for which no stable pair of schedules exists. Let a_1, a_2 be set the shows of A and let b_1; b_2 be the set of shows of B. Let the ratings of a_1; a_2; b_1; b_2 be 1, 3, 2 and 4 respectively. In every schedule that A and B can choose, either a_1 shares a slot with b_1 or a_1 shares a slot with b_2. If a_1 shares a slot with b_1, then A has an incentive to make a_1 share a slot with b_2 thereby increasing its number of winning slots from 0 to 1. If a_1 shares a slot with b_2, then B has an incentive to make b_2 share a slot with a_2 thereby increasing its number of winning slots from 1 to 2. Thus every schedule that A and B can choose is unstable.

   Rubric (6pt):

   - 1 pt: Correct claim that there are configurations of shows without stable matching
   - 5 pt: Provides a correct counterexample

7. Solve Kleinberg and Tardos, Chapter 1, Exercise 4.

   We will use a variation of Gale and Shapley (GS) algorithm, then show that the solution returned by this algorithm is a stable matching.

   In the following algorithm (see next page), we use hospitals in the place of men; and students in the place of women, with respect to the earlier version of the GS algorithm given in Chapter 1.

   This algorithm terminates in $O(mn)$ steps because each hospital offers a position to a student at most once, and in each iteration some hospital offers a position to some student.

   The algorithm terminates by producing a matching M for any given preference list. Suppose there are $p > 0$ positions available at hospital h. The algorithm terminates with all of the positions filled, since, any hospital that did not fill all of its positions must have offered them to every student. Every student who rejected must be committed to some other hospital. Thus, if h still has available positions, it would mean total number of available positions is $> n$, the number of students. This contradicts the assumption given, proving that all the positions get filled.

```
1:  while there exists a hospital h that has available positions do
2:    Offer position to the next highest ranked student s in the preference list
      of h that wasn't offered by h before
3:    if s has not already accepted a position at another hospital h' then
4:      s accepts the position at h.
5:    else
6:      Let s be matched with h'.
7:      if s prefers h' to h then
8:        then s rejects the offer of h
9:      else
10:       s accepts the position at h freeing up a position at h'
11:     end if
12:   end if
13: end while
```

The assignment is stable. Suppose, towards a contradiction, that the M produced by our adapted GS algorithm contains one or more instabilities. If the instability was of the first type (a student s' was preferred over a student s by a hospital h but was not admitted), then h must have considered (offered) s before s' who wasn't offered, which is a contradiction because h prefers s' to s. Thus, the instability was not of the first type. If the instability was of the second type (there is student s, s' currently at hospitals h, h' and there's a swap mutually beneficial to h and s'), then h must not have admitted s' when it considered it before s, which implies that s' prefers h' to h, a contradiction. Thus, the instability was not of the second type. Thus, the matching was stable. Thus at least one stable matching always exists (and it is produced by the adapted GS algorithm as above).

Rubric (15pt):

- 8pts: Algorithm
  (a) 1pt: Loop condition (line 1)
  (b) 2 pt: hospitals offer next highest ranked student (line 2)
  (c) 2pts: case that s is free (lines 3-4)
  (d) 3pts: cases if s is at another h'
- 7pts: Proof
  (a) 1 pt: Algorithm terminates in finite steps (optional to mention in O(mn) steps)
  (b) 2pt: All positions get filled
  (c) 2pts: Explain why no instability of first type
  (d) 2pts: Explain why no instability of second type

8. N men and N women were participating in a stable matching process in a small town named Walnut Grove. A stable matching was found after

the matching process finished and everyone got engaged. However, a man named Almazo Wilder, who is engaged with a woman named Nelly Oleson, suddenly changes his mind by preferring another woman named Laura Ingles, who was originally ranked right below Nelly in his preference list, therefore Laura and Nelly swapped their positions in Almanzos preference list. Your job now is to find a new matching for all of these people and to take into account the new preference of Almanzo, but you don't want to run the whole process from the beginning again, and want to take advantage of the results you currently have from the previous matching. Describe your algorithm for this problem.
Assume that no woman gets offended if she got refused and then gets proposed by the same person again.

First, Almanzo leaves Nelly and proposes to Laura:

- If Laura rejects Almanzo and decides to stay with her current fiance (her current fianc is ranked higher than Almanzo in her preference list), then Almanzo just goes back and proposes to Nelly again and gets engaged with her. Algorithm stops. This is where you take advantage of the previous matching.
- Else, Laura accepts Almanzo proposal, she will get engaged with him and leave her current fiance (say Adam). And here comes the main part of the matching puzzle.
  (a) Note that Adam may or may not simply propose to Nelly and be engaged with her. The reason is Nelly may be far below Laura in his preference list and there are more women (worse than Laura and better than Nelly), to whom he prefers to propose first. Furthermore, the fact that Nelly becomes single can create more instabilities for those men, who once proposed to Nelly and were rejected by her because she preferred Almanzo. Thus, Nelly is ranked higher than the current fiances of those once unlucky men and they are ready to propose to Nelly again for a second chance.
  (b) Thus, one way is to put Adam and all those rejected-by-Nelly men in the pool of single men, and put Nelly and the fiancees of the rejected-by-Nelly men in the pool of free women. Similar to Nellys case, moving any woman from the engaged state to the single state may create more instabilities; thus one needs execute step (b) recursively to build the pools of free men and women. In the worst case, all men and all women will end up in the single pools; in the best case (when Laura accepted Almanzo's proposal), only Adam and Nelly are in the 2 single pools, one for men and another for women.
  (c) Execute G-S algorithm until there is no free man. For each man, if there is a woman who once rejected him and is now free, he will try to propose to her again. Otherwise, he will propose to

those women that he has never proposed to, moving top down in his preference list. In the latter case, more men (subsequently more women) may go to the single pools.

Note: The above solution is described in details in order to clarify the problem. For this problem, students can simply write pseudo code without having to discuss the details of each case as above.

Rubric (15pt):

- 2pt: Cover the early termination case (if Laura rejects Almanzo, the matching stays as is).
- 2pts: Mention the instability of Adam once Laura leaves him
- 3pts: Mention the instability of men that were rejected by Nelly.
- 5pts: Recursively building the pools of free men and women
- 3pts: Executing GS **while exapanding single pools as needed at each step**.

Completely acceptable if any of these points is not explicitly described but the algo/pseudocode covers it.