Name: Chaoyu Li

USC ID: 6641732094

Date: 8/28/2021

CSCI 570 - Fall 2021 - HW 1

Due: Sep 3rd, 11.59 PM PST

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

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

False.

Let us suppose there are three men m, m1 and three women w, w1. The order of their preferences is as follows:

Men:

| m | w | w1 |
| --- | --- | --- |
| m1 | w1 | w |

Women:

| w | m1 | m |
| --- | --- | --- |
| w1 | m | m1 |

The only two possible stable matching are {(m,w),(m1,w1)} and {(m,w1),(m1,w)}. Neither of them meet the requirement on the claim in Exercise 1.

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

True.

Let us suppose, by way of contradiction, there is a stable matching set S1 does not contain a pair (m,w), and w is the highest rank on m’s list, m is the highest rank on w’s list. Consider in this situation, there are two pairs (m,w1) and (m1,w), then (m,w) become an unstable pair because for m:w>w1 and for w:m>m1. Thus, S1 is not stable and this is a contradiction.

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.

Let us suppose that a stable matching set S1 is obtained from the version where the male proposes, then according to the theorem ‘The active get best and the passive get worst,’there is a conclusion that every man in this set gets their best valid partner and every woman gets their worst valid partner. Then we suppose that another stable matching set S2 is obtained from the version where the female proposes, then every woman in this set gets their best valid partner and every man gets their worst valid partner. We assume that S1=S2, then we get that for every person, their best valid partner is the same as their worst valid partner. This implies that every person has a unique valid partner which implies that there is a unique stable matching.

Let us suppose that the version of male proposes and the version of female proposes both yield the exact same matching S. For every man in S, he gets his best valid partner in the version of male proposes and he gets his worst valid partner in the version of female proposes. It means his best valid partner and worst valid partner is the same woman. Thus, there is only one stable choice for everyone, the stable matching is unique.

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 does not always exist.

Let us suppose an example as follow:

| a | b | c | d |
| --- | --- | --- | --- |
| b | c | a | d |
| c | a | b | d |
| d | a | b | c |

In this case, because d is the last choice for a,b,c, no matter what the combination is, d's roommate must be dissatisfied. At the same time, because a like b most, b like c most, and c like a most, a person in another room must be dissatisfied either. So the matching is unstable in every combination.

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

A stable schedule does not always exist.

Let us suppose that Network A has two TV Shows {a1,a2} with rating {20,40}, and Network B has two TV Shows{b1,b2} with rating {10,30}. In this case, Network A has a chance to win two schedules and Network B has a chance to win one schedule. Network B will always prefer to let b2 against with a1, but Network A will prefer to let a2 against with b2. Thus, there will never be a stable schedule.

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

Algorithm:

While still has a hospital hi that has position:

hi send offer to student sj (sj is the highest rank in hi’s current list)

if sj is free:

Accept the offer

else:

if sj prefer hi to hj: (hj is the current hospital that sj accept)

sj accept hi’s offer

hi’s position - 1

hj’s position + 1

else:

sj refuse hi’s offer

Proof:

The time complexity of this algorithm is O(nm) because each hospital will send an offer to each student once at most.

1. For the first type of instability:
   1. Let us suppose that student s is assigned to hospital h, s’ is assigned to no hospital, and h prefers s’ to s.
   2. According to the algorithm above, h will send offers according to the rank list, so they will send an offer to s’ before s.
   3. If s’ is free that time, h will hire s’ instead of s.
   4. If s’ refuses h’s offer, then he must have a better offer from another hospital. Then s’ will not be assigned to no hospital in the end.
   5. Thus, this is a contradiction.
2. For the second type of instability:
   1. Let us suppose there are two pairs (s,h) and (s’,h’). In addition, for s: h’>h; for h’:s>s’.
   2. According to the algorithm above, if h’ have available position, it will send an offer to s before s’.
   3. If s has a better offer from another hospital that time, he will refuse the offer from h’, AND he will also refuse the offer from h because for s: h’>h. Otherwise, s will accept the offer from h’ and refuse the offer from h.
   4. So the pair will alway be (s,h’) instead of (s,h).
   5. Thus this is a contradiction.

**According to the proof, the algorithm above is stable.**

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 Almanzo 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.

Let us assume the man that Laura is matching named B.

Algorithm:

If B>Almanzo in Laura’s list:

Laura refuses Almanzo’s propose

else:

Laura accepts Almanzo’s propose

Nelly and B become free.

Change the status of all men whose rank is between Almanzo and B in Nelly’s list to be free

Change the status of all women who is matching with these men to be free

Change the status of all women whose rank is between Laura and Nelly in Almanzo’s list to be free

Change the status of all men who is matching with these women to be free

Run G-S algorithm again for all free men and women

Let us suppose that the man who match with Laura named B.

Algorithm:

If B>Almanzo in Laura’s list:

Laura refuses Almanzo’s propose

Almanzo return to Nelly

Else:

Laura accepts Almanzo’s propose

Nelly and B become free, it may cause some instabilities

Put B and all men whose rank is between Almanzo and B in Nelly’s list into the single pool of men

Put all women whose ficanee has been put into the single pool of men into the single pool of women, it may create more instabilities

***Recursively execute the two steps above until there is no more instability***

Run G-S algorithm again for all free men and women