Assignment 1

Ayman Shahriar UCID: 10180260

September 18, 2020

Question 1

Precondition:

- A set M = $\{m_1, \ldots, m_n\}$ of n men
- A set W = $\{w_1, \ldots, w_n\}$ of n women
- For each man, a ranking (ordered list) of all women in order of preference
- For each woman, a ranking (ordered list) of all men in order of preference

Postcondition:

- Return a perfect matching S of pairs $(m, w) \in (M \times W)$ such that S is a stable matching

Reference: Slide 11, set of notes titled "Topic 1"

Question 2

This statement is false. We will show this with an example:

- $M = \{m_1, m_2\}$
- $W = \{w_1, w_2\}$
- $\begin{array}{ll} m_1 : \ (w_1, w_2) & w_1 : \ (m_2, m_1) \\ m_2 : \ (w_2, w_1) & w_2 : \ (m_1, m_2) \end{array}$

In this instance we can have 2 possible perfect stable matchings:

$$S_1 = \{(m_1, w_1),\ (m_2, w_2)\}$$

$$S_2 = \{(m_1, w_2),\ (m_2, w_1)\}$$

In S_1, both man are with their preferred partners, so they have no incentive to leave, which makes the matching stable. And in S_2 both women are with their preferred partners, so they have no incentive to leave, which makes all the matching stable.
So in this instance, none of the perfect stable matchings contain a pair where the man and woman ranked each other first, so this statement is false.

Question 3

This statement is true.

That is because if m is paired up with w' and w is paired up with m' (where $m \neq m'$, $w \neq w'$), then m and w will prefer each other over their current partners since m ranked w over w' and w ranked m over m', which means that (m, w) will form an instability.

So the only way to make the perfect matching stable is to include the pairs (m, w).

Question 4

The best case will occur if all the men rank a different woman first in their preference list (the preferences lists of the women don't matter). In the best case, the loop will execute n times.

This is because the condition of the while loop to execute is that there has to be at least one unpaired man who hasn't proposed yet, so the only way for the loop to end is if all the men become paired.

And in order for a man to become paired, he must propose at least once. Since each iteration of the loop involves a single man proposing, the lowest number of times the while loop can execute is n times. This will occur in cases where each man ranks a different woman first on their list. In those cases, there is no possibility of a woman leaving her current partner for another, so each man will have to propose exactly once throughout the execution of the algorithm (and so the loop will execute n times).

The worst case will occur when the preference lists of all the men are exactly the same (the preference lists of the women do not matter). In the worst case, the loop will execute at most n^2 times.

Each time the loop executes, a man proposes to a woman that he has not proposed to before. So in each iteration of the loop, the number of pairs $(m, w) \in (M \times W)$ where m proposed to w increases by 1, so we can use this as a measure of progress for our algorithm.

However, since there are only $n \times n = n^2$ possible pairs of men and women, it means that the loop can execute at most n^2 times before the algorithm terminates.

In the case where the preference lists of all the men are the same, every man will propose to the highest ranked woman w_1, and the man m_1 that w_1 ranked highest will be permanently paired with w_1. Then the remaining men will propose to the second highest ranked woman w_2, and the man m_2 that w_2 ranked highest will be permanently paired with w_2. This will continue until we are left with only one remaining man proposing to his last ranked woman, and they become paired.

So every unpaired man will propose to as many women as possible before they get paired permanently, which means the number of times the loop executes will be close to the upper bound.

Reference: Page 7 of the textbook

Question 5

No, for every set of shows and ratings, there is not always a stable pair of schedules. We will prove this with an example.

Let $n = 2$. So both networks A and B will have two shows and two time slots each.

Assume that a show's rating can be any integer. Let the two shows of A be rated 0 and 100. Let the two shows of B be rated 1 and 101

Then we will have two possible pairs of schedules (assume the order of the time slots do not matter):

Pair 1:

	Slot 1	Slot 2
A	0	100
B	101	1

Pair 2:

	Slot 1	Slot 2
A	0	100
B	1	101

In Pair 1, A will win Slot 2 and B will win Slot 1. But if B ends up switching it's shows, then we will end up with Pair 2 where B will win both time slots.

However A can then switch it's shows so that we go back to pair 1 where A wins a time slot, so we are caught in a loop where each network can change their schedule to win more time slots. Thus, there is no stable pair of schedules for this example.