CMSC 27200 - Winter 2021 — Andrew Drucker, Lorenzo Orecchia
due tuesday january 19 at 12 noon via gradescope

H1.hamlet

- The assignment is due at Gradescope on Tuesday, January 19 at 12:00 noon.
- You can either type your homework using LaTex or scan your handwritten work. We will provide a LaTex template for each homework. If you writing by hand, please fill in the solutions in this template, inserting additional sheets as necessary. This will facilitate the grading.
- You are permitted to study with up to 2 other students in the class and discuss the problems; however, *you must write up your own solutions, in your own words*. Do not submit anything you cannot explain. If you do collaborate with any of the other students on any problem, please do list all your collaborators in your submission for each problem.
- Similarly, please list any other source you have used for each problem, including other textbooks or websites. *Consulting problem solutions on the web is not allowed.*
- *Show your work.* Answers without justification will be given little credit.

PROBLEM 1 (25 POINTS) *Answer the questions below using the following preference lists:*

Group A's preference lists (from most preferred to least preferred):

a_1: b_1, b_4, b_3, b_2

a_2: b_4, b_3, b_1, b_2

a_3: b_1, b_2, b_4, b_3

a_4: b_3, b_1, b_2, b_4

Group B's preference lists (from most preferred to least preferred):

b_1: a_4, a_2, a_3, a_1

b_2: a_1, a_4, a_2, a_3

b_3: a_3, a_1, a_4, a_2

b_4: a_4, a_3, a_1, a_2

(a) Run the Gale-Shapley algorithm with group A making the offers to obtain a stable matching. For your answer (and for your answer to part b as well), at each step please give the offer that is made and whether or not this offer is accepted. You should also give the final matching you obtain.

(b) Now run the Gale-Shapley algorithm with group B making the offers to obtain another stable matching. Which people are happier in this new stable matching (compared to the stable matching found in part a)?

(c) What other stable matching(s) are there, if any? Note: For full credit you should show that you have indeed found all of the possible stable matchings.

Solution: Collaborators: Yael Sulkin

Part A:

OFFER	ACCEPT?	COMMENTS
a_1 to b_1	Yes	a_1, b_1 ENGAGED
a_2 to b_4	Yes	a_2, b_4 ENGAGED
a_3 to b_1	Yes	a_3, b_1 ENGAGED, a_1 is now FREE
a_1 to b_4	Yes	a_1, b_4 ENGAGED, a_2 is now FREE
a_2 to b_3	Yes	a_2, b_3 ENGAGED
a_4 to b_3	Yes	a_4, b_3 ENGAGED, a_2 is now FREE
a_2 to b_1	Yes	a_2, b_1 ENGAGED, a_3 is now FREE
a_3 to b_2	Yes	a_3, b_2 ENGAGED

FINAL PAIRINGS: (a_1, b_4) (a_2, b_1) (a_3, b_2) (a_4, b_3)

Part B:

OFFER	ACCEPT?	COMMENTS
b_1 to a_4	Yes	b_1, a_4 ENGAGED
b_2 to a_1	Yes	b_2, a_1 ENGAGED
b_3 to a_3	Yes	b_3, a_3 ENGAGED
b_4 to a_4	No	b_4 is still FREE
b_4 to a_3	Yes	b_4, a_3 ENGAGED, b_3 is now FREE
b_3 to a_1	Yes	b_3, a_1 ENGAGED, b_2 is now FREE
b_2 to a_4	No	b_2 is still FREE
b_2 to a_2	Yes	b_2, a_2 ENGAGED

FINAL PAIRINGS: (b_1, a_4) (b_2, a_2) (b_3, a_1) (b_4, a_3)

HAPPIER: b_1, b_2, b_3, b_4 (they were able to choose).

Part C: The pairings found in A and the pairings found in B are all the stable pairings. This can be proved using the fact that the G-S algorithm finds proposer's best valid partner and proposee's worse possible partner. Thus when looking at any (a,b) all the stable matching that can be found lie in between the overlap. Let's use a_1 as an example. In the answer to Part A we found that since a did the proposing, the best valid partner is b_4, which was second on a_1's ranking. In the answer to Part B, we found that when a was ebing proposed to, the worst valid partner is b_3, which was third on a_1's list. Since b_4 was the best, and b_3 was the worst then any rankings in between would be possible stable macthings. However, since there existed no b's ranked in between b_4 and b_3, (a_1, b_4) and (a_1, b_3) are the only stable matching for a_1. This proof holds for all a's and b's, where the stable matchings already found are in fact all the stable matchings that exist for this set of preferences.

Group A's preference lists (best and worst):

a_1: [b_4 BEST, b_3 WORST]

a_2: [b_1 BEST, b_2 WORST]

a_3: [b_2 BEST, b_4 WORST]

a_4: [b_3 BEST, b_1 WORST]

Group B's preference lists (best and worst):

b_1: [a_4 BEST, a_2 WORST]

b_2: [a_2 BEST, a_3 WORST]

b_3: [a_1 BEST, a_4 WORST]

b_4: [a_3 BEST, a_1 WORST]

Extra Space for your solution

PROBLEM 2 (30 POINTS) *Solve exercise 8 in Chapter 1 in the Kleinberg-Tardos textbook.*

Solution: Collaborators: Yael Sulkin

Give a proof that, for any set of preference lists, switching the order of a pair on the list cannot improve a woman's partner in the GaleShapley algorithm

Proof For this proof let's assume we have three men, m, m', and m'' and three women, w, w', w''. Let's also assume w: $m'' > m > m'$ but switched the order of m and m' so that the G-S Algorithm works with w: $m'' > m' > m$. Our goal for this proof is to show that while w always prefers m'', dishonestly switching the order of lesser preferences does not benefit the person. We will work backwards.

For w to be engaged with m'' there are two scenarios: either both are free and m'' proposes to w or w is already engaged, m'' is free, and m'' has not already proposed to w. In the first scenario, the order of lesser preferences for w does not impact her happiness because w would be engaged to her top choice and according to statement 1.1 of the KT textbook, a woman's partner will never get worse. In fact, in any case where m'' proposes w will be happiest. Whether they are engaged to m or m', a proposal from m'' will result in (m'',w). In the case m'' never porposes, if w is engaged to m and m' proposes according to the G-S algorithm, w will leave and pair with m'. To the algorithm this is w getting a better matching but according to their true preferences w is getting a worse pairing. In fact, because the G-S algorithm enables women to have better and better matchings, any w for any set of preferences would be best off staying true to their rankings. Switching lesser preferences does not impact the chances of getting their first choice but it runs the risk of w pairing with her worse option rather than a better one as long as the worse option is ranked higher.

For a switch in a pair of preferences to be a direct cause of improvement in their matching, we will have to run the G-S Algroithm twice, once before the switch $w : m'' > m > m'$ and once after the switch $w : m'' > m' > m$. To be considered an improvement, the result of the algorithm with w's true preferences must be (m', w) and the result with w's false preferences must be (m, w). With two runs of the G-S algorithm and two proposers, we have 4 ways the G-S algorithm could run: m proposes to w first and m' second, or m' proposes to w first or m second. Below is a table of each possible scenario:

RUN	WORKING PEFERENCE LIST	PROPOSER 1	RESULT	PROPOSER 2	FINAL RESULT
1	$w : m'' > m > m'$ (TRUE)	m	(w, m)	m'	(w, m)
	$w : m'' > m' > m$ (FALSE)	m	(w, m)	m'	(w, m')
2	$w : m'' > m > m'$ (TRUE)	m'	(w, m')	m	(w, m)
	$w : m'' > m' > m$ (FALSE)	m'	(w, m')	m	(w, m')

There is no improvement. In all cases in which w switches a pair of her preferences, it does not directly cause w to be matched with someone they prefer.

Extra Space for your solution

PROBLEM 3 (30 POINTS) *Suppose you are given two distinct stable matchings P and Q of n hospitals to n students. Construct a new matching R using the following rule: For each hospital h who is matched to two different students s and s' in P and Q, h matches to its preferred student between s and s' in R.*

1. *Show that R is indeed a matching.*
2. *Show that R is stable.*

Solution:

Proof to 1 We can prove by contradiction. Let's assume h: $s > s'$ since the proof can work vice versa. We assume (h, s) or (h, s') is not a match. However, this contradicts the given that (h, s) exists as a stable match in P and (h, s') exists as a stable match in Q. Since P is a list of stable matchings, we know that there exists a perfect matching between hospitals and students by definiton. As in, every hospital matches with one student only and every student matches with one hospital only: a one-to-one relationship. Thus, as long as there exists an h not paired with a student, and students s, s' in hospital h's preference list, there exists a match in R.

Proof to 2 We can prove by contradiction. Let's assume either relationship (h, s), (h, s') is not stable. This proof can work for either option but let's work with (h, s) which is in P. If (h, s) in R is not stable then there exists some h' in R such that h' prefers s to s' and s prefers h' to h. This would result in an instability such that (h', s) is a better match than (h, s). However, this contradicts the fact that (h, s) was a stable match in P. Since we are given that (h, s) was stable, we know there does not exist an h' such that (h', s) is a better match than (h, s) in R.

Extra Space for your solution

PROBLEM 4 (15 POINTS) *Complete the following review exercises about big-O notation.*

(a) *Describe each of the following functions $T : \mathbb{N} \to \mathbb{R}^+$ using big O notation. For full credit, the big-O expression should be as simple as possible (to a reasonable observer). Note: For this problem it is sufficient to describe the upper bound for each of these functions. For example, if we had $T(n) = 10000$, we could say that $T(n)$ is $\Theta(1)$ but it is sufficient to say that $T(n)$ is $O(1)$.*

1. $T(n) = 5n \log_2(n) + 2n + 3$
2. $T(n) = 20n + n^2$
3. $T(n) = 100 + 2\sqrt{n} + 7\log_2(n)$
5. $T(n) = 10n^2 \cdot 4^n$

(b) *Prove that for functions $f, g : \mathbb{N} \to \mathbb{R}^+$, $f(n)$ is $O(g(n))$ if and only if*

$$\exists n_0, C : \forall n \geq n_0, \log(f(n)) \leq \log(g(n)) + C$$

(c) *Order the following bounds from smallest to largest: $O(n^2)$, $O(2^n)$, $O(1)$, $O(\log_2(n))$, $O(n\log_2(n))$, $O(\sqrt{n})$, $O(n^{\log_2(n)})$, $O(n)$, $O(n^{\sqrt{n}})$, $O((\log_2(n))^{10})$. [**Hint:** Use part (b)]*

Solution: Collaborators: Yael Sulkin

Part A:

1. $T(n) = 5n\log_2(n) + 2n + 3 \Rightarrow T(n) = O(nlog(n))$
2. $T(n) = 20n + n^2 \Rightarrow T(n) = O(n^2)$
3. $T(n) = 100 + 2\sqrt{n} + 7\log_2(n) \Rightarrow T(n) = O(\sqrt{n})$
5. $T(n) = 10n^2 \cdot 4^n \Rightarrow T(n) = O(4^n)$

Part B:

Proof if $f, g : \mathbb{N} \to \mathbb{R}^+$, $f(n) = O(g(n))$ then we know there exists some constant c such that $f(n) \leq g(n) * c$. If we take the logarithm of both sides we get $log(f(n)) \leq log(g(n) * c) \Rightarrow log(f(n)) \leq log(g(n) + log(c))$. Since c is a constant, log(c) is also a constant value which we can denote with C. Therefore, $log(f(n)) \leq log(g(n)) + C$.

Part C: $O(1) < O(log(n)) < O(\sqrt{n}) < O(n) < O(nlog(n)) < O((log(n))^{10}) < O(n^{log(n)}) < O(n^2) < O(n^{\sqrt{n}}) < O(2^n)$

Extra Space for your solution