EECS 70 Discrete Mathematics and Probability Theory
Fall 2014 Anant Sahai

Homework 3

This homework is due September 22, 2014, at 12:00 noon.

1. Propose-and-Reject Lab

In this week's Virtual Lab, we will simulate the traditional propose-and-reject algorithm. You will find a simple implementation of a `Person` class, where each `Person` has an identification number, is either male or female, and has his or her own preference list, in the code skeleton for this question.

Please download the IPython starter code from Piazza or the course webpage, and answer the following questions.

(a) Using the example from Note 4, page 6, which is shown again below for your convenience, create one list which contains all the men, and one which contains all the women. Each man or woman needs to be created using the `Person` class.

Men	Women			
1	A	B	C	D
2	A	D	C	B
3	A	C	B	D
4	A	B	C	D

Women	Men			
A	1	3	2	4
B	4	3	2	1
C	2	3	1	4
D	3	4	2	1

(b) For this question, we've given you the code to run the traditional propose-and-reject algorithm where the men propose to the women. Your task is to create a new version by changing it to women making proposals. Are the final pairings for the two cases the same?

Solutions: They are not in this case.

(c) This question will focus on exploring what happens when we randomize preferences. In the code skeleton, you will find an implementation of the function `create_random_lists`, which creates a random set of preferences.

We're interested in how often the men-propose and women-propose algorithms return the same pairings. Write a function that generates a random set of preferences for men and women, then runs each variant of the traditional propose-and-reject algorithm on that set of preferences. For lists of 4 people, how often do the men-propose and women-propose algorithms agree on the final stable pairings?

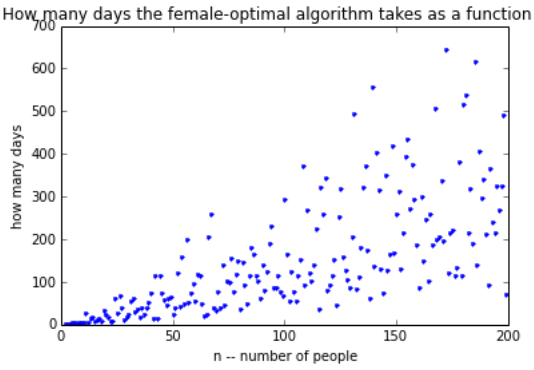
Solutions: The value of `same_final_pairing(n,k)` is usually around $\frac{2n}{k}$ for sufficiently high values of n and k. Intuitively, this makes sense. Why? n is like our sample size - since each set of preferences is independent from the last. As we increase n, the value should increase linearly. Furthermore, we'd expect higher values of k to decrease the frequency of both pairings being the same, since there are more possible pairings and thus it's less likely for the female-optimal and male-optimal pairings to be the same.

(d) Finally, we're going to explore how long the women-propose algorithm takes as a function of n. In lecture, we learn that the algorithm must terminate after at most n^2 days (we will soon prove a stricter bound in a later question). Write a function that returns how many days it takes for the traditional propose-and-reject algorithm to arrive at its stable solution using randomly-generated preference lists.

Try it on some inputs of different size. How quickly does the number of days grow with the input? Does it grow linearly (at the same rate as n - maybe twice as fast or half as fast), quadratically, loga-

rithmically? If you can't tell, try graphing some outputs by hand. You do not have to submit any graph, but it certainly would help defend your claim.

Solutions: The value appears to grow approximately linearly. The most important thing to note here is that the upper bound of n^2 is virtually never even approached. Usually, even for large n, the highest value seen is less than $3n$. Distinguishing between linear and loglinear ($nlog(n)$) is very difficult to do empirically, especially for small n like we have here, so both answers are acceptable.



Reminder: When you finish, don't forget to convert the notebook to pdf and merge it with your written homework. Please also zip the `ipynb` file and submit it as `hw3.zip`.

2. Candy Problem

There are N students standing in a circle, facing the center. Each student is initially issued an even number of candies. Being the fair and honest students that they are, they come up with the following adjustment algorithm. First, each student gives half of his or her candies to the student on his or her left. Note that after this step, some students might have an odd number of candies. Next, those students with an odd number of candies will get one more candy from the teacher. The students repeat the adjustment algorithm and stop when everyone has the same number of candies.

(a) Run the algorithm for the case where there are 6 students and the initial number of candies, in clockwise order, is: {2, 4, 4, 2, 6, 8}.

Solutions: Initial configuration: $\{2,4,4,2,6,8\}$
After iteration 1: $\{6,4,4,4,4,8\}$
After iteration 2: $\{8,6,4,4,4,6\}$
After iteration 3: $\{8,8,6,4,4,6\}$
After iteration 4: $\{8,8,8,6,4,6\}$
After iteration 5: $\{8,8,8,8,6,6\}$
After iteration 6: $\{8,8,8,8,8,6\}$
After iteration 7: $\{8,8,8,8,8,8\}$

(b) By the Well-Ordering Principle, before we begin the adjustment algorithm, there is a minimum number of candies possessed by any student and a maximum number of candies possessed by any student. Since these quantities are even, we let $2n$ denote the minimum number of candies and let $2m$ denote the maximum number of candies in the initial candy distribution. Prove that the number of candies possessed by each student is still between $2n$ and $2m$ after one iteration of the adjustment algorithm.

Solutions: Let s be any student, and let $2k$ be the number of candies that s has. Let t be the student to the right of s, and let $2r$ be the number of candies that t has. We know that at the beginning of the algorithm, $n \leq k, r \leq m$. After the first step of the adjustment algorithm, student s will have $k+r$ candies. In the second step of the adjustment algorithm, there are two cases:

- If $k+r$ is even, then s is not given an additional candy, so s will have $k+r$ candies after the first iteration. Since $n \leq k, r \leq m$, we know that $2n \leq k+r \leq 2m$.
- If $k+r$ is odd, then s will be given an additional candy, and so s will have $k+r+1$ candies after the first iteration. Since $k+r$ is odd, we can't have that both $k=m$ and $r=m$, since in this case $k+r=2m$ would be even. Thus we must have that at least one of k, r is strictly less than m, so $2n \leq k+r < 2m$, and therefore $2n \leq k+r+1 \leq 2m$.

Since the number of candies that s has is still between $2n$ and $2m$ for both cases, and we could have picked any of the students as s, the result holds.

(c) Now, suppose the minimum number of candies that any student has immediately after iteration i is $2k$, and that p students have exactly $2k$ candies. Prove that, provided the algorithm doesn't terminate immediately after iteration i, less than p students will have exactly $2k$ candies immediately after iteration $i+1$.

Solutions: To prove this, we will prove that the two claims hold: (1) Students with more than $2k$ candies still have more than $2k$ candies after one iteration, and (2) At least 1 out of the p students will have a larger number of candies after one iteration.

(1) Assume student s has 2ℓ candies where $\ell > k$. On his or her right, there is a student with $2r$ candies where $r \geq k$. After one iteration, student s has $\ell + r$ candies. Since $\ell + r > k + r \geq 2k$, student s still has more than $2k$ candies after one iteration.

(2) First, we will prove that at least 1 of the p students has someone with more than $2k$ candies on his or her right.

Proof by contradiction: Assume no student with $2k$ candies has someone with more than $2k$ candies on his or her right. Then all the students have $2k$ candies, which means the algorithm would have terminated. Contradiction!

From the proof above, we can assume that student s with $2k$ candies has someone with $2r$ candies on his or her right where $r > k$. After one iteration, student s has $k+r$ candies. As $k+r > k+k = 2k$, s now has more than $2k$ candies. Therefore, at least 1 out of p students will have a larger number of candies after one iteration.

From (1), p cannot increase, since the students who have more than $2k$ candies will still have more than $2k$ candies after one iteration. From (2), at least one student who had $2k$ candies will have more than $2k$ candies after one iteration. Therefore, p must decrease by at least 1.

(d) Does the adjustment algorithm always terminate in a finite number of iterations, or could students be trading candies with each other forever? Explain your reasoning.

Solutions: The algorithm will terminate in a finite number of steps. The intuition is that we have proved in (b) that the maximum number of candies is bounded and we have proved in (c) that the minimum number of candies will increase, so the minimum number of candies will approach the maximum number of candies until the final iteration where these quantities are the same.

Assume that the maximum number of candies is $2m$ and the minimum number of candies is $2n$ where n and m are some integers and $n \leq m$ in the initial configuration. If $n = m$, the statement holds since everyone starts with the same number of candies and the algorithm terminates after 1 step. Now, consider the case that $n < m$:

Let p be number of students with the minimum number of candies in the initial configuration. From (c), at least 1 student with the minimum number of candies will have more candies after running one iteration. Therefore, all the students with the minimum number of candies in the initial configuration have more than $2n$ candies after at most p iterations. Repeating the algorithm, the minimum number of candies must increase within a finite number of iterations, while by (b), the maximum number of

candies cannot increase. So the minimum number of candies must be equal to the maximum number of candies in a finite number of steps.

3. Indifferent Attitudes

In the real world, it is perhaps a bit unrealistic to expect that each man and woman can provide a *strict* preference ordering of all his or her potential mates. To reflect this, let's suppose we allow each person's preference list to contain *ties*. Mathematically, a set W of k women forms a tie of length k in the preference list of man m if m does not prefer w_i to w_j for any $w_i, w_j \in W$ (i.e., m is *indifferent* between w_i and w_j), while for any other woman w not in W, either m prefers w to all women in W or m prefers all the women in W to w. A tie on a woman's list is defined analogously. We will call this the *Stable Marriage Problem with Ties*.

Recall that in the original Stable Marriage Problem, where preference lists are strictly ordered, it is always possible to find a stable matching, where a matching is stable if there is no man x and woman y such that x and y both prefer each other over their current partners. However, for the Stable Marriage Problem with Ties, 3 different types of stability are possible:

- **Weak stability.** A matching is *weakly stable* if there is no couple x and y, each of whom strictly prefers the other to their current partner in the matching.
- **Strong stability.** A matching is *strongly stable* if there is no couple x and y such that x strictly prefers y to his or her partner, and y either strictly prefers x to his or her partner or is indifferent between them.
- **Super-stability.** A matching is *super-stable* if there is no couple x and y, each of whom either strictly prefers the other to his/her partner or is indifferent between them.

It is then natural to ask whether these different types of stable matchings always exist in a given instance of the Stable Marriage Problem with Ties. Answer the following.

(a) For the Stable Marriage Problem with Ties, does a weakly stable matching always exist? Either prove the statement or provide a counterexample.

Solutions: It is easy to see that a weakly stable matching always exists. We will take our instance of the Stable Marriage Problem with Ties and reduce it to an instance of the usual Stable Marriage Problem with strict preferences. To do this, we will break ties arbitrarily. For instance, we could simply say that partners that are tied are actually preferred based on a combination of alphabetical order — so Alice is preferred to Betty if they are tied — and age: so if a woman has tied preferences for Aaron's, she'll prefer the older one. How we break ties doesn't matter at all. This is just one way that tie-breaking could happen.

Then if we run the usual propose-and-reject algorithm on our constructed instance with strict preferences, the stable matching produced by the algorithm is a weakly stable matching for our original instance of the Stable Marriage Problem with Ties. In fact, any matching that is stable in the constructed instance (preference lists with ties broken) is weakly stable in the original instance (with ties). We will prove this by contradiction.

Let M be any instance of the stable marriage problem with ties. Break ties in the preference lists in M arbitrarily to create an instance of the stable marriage problem (with strict preference orders), M'.

Assume that there exists some matching L, where L is a stable matching for M' but L is not a weakly stable matching for M. Since L is not a weakly stable matching for M, then by the definition of weak stability, there exist couples $(m, w'), (m', w) \in L$ such that in the preference lists in M, m strictly prefers w to his current match w' and w strictly prefers man $m$to her current match m'. But then (m, w) would also forms a rogue pair in L for instance M'. But this contradicts our assumption that L is a stable matching for M'. Thus our assumption must be false.

(b) For the Stable Marriage Problem with Ties, does a strongly stable matching always exist? Either prove the statement or provide a counterexample.

Solutions: It is not always possible to find a strongly stable matching. Here is one possible counterexample:

Man	Preference List
1	$A > B$
2	$A = B$

Woman	Preference List
A	$2 > 1$
B	$2 > 1$

Since both women A and B prefer man 2, and man 2 is indifferent to both women A and B, both possible matchings $\{(1,A),(2,B)\}$ and $\{(1,B),(2,A)\}$ are not strongly stable.

(c) For the Stable Marriage Problem with Ties, does a super-stable matching always exist? Either prove the statement or provide a counterexample.

Solutions: It is not always possible to find a super-stable matching. An easy counter example is one where everyone is indifferent to all their potential partners. You could have also reused your solution to part (b) above, since if an instance is not strongly stable, it is also not super stable.

(d) Assume we are given an instance of the Stable Marriage Problem with Ties, along with a weakly stable matching M for that instance. Upon getting married to their partners assigned in M, each person's preferences change slightly and the married partner becomes preferred over anyone they were tied with, but doesn't change in rank otherwise. Is M now super-stable?

Solutions: Yes. M is now super-stable.

Let P be our original instance of the Stable Marriage Problem with Ties, and let M be a weakly stable matching for P. Let P' be the modified instance of the Stable Marriage Problem with Ties (i.e., where the married partner in M becomes preferred over anyone they were tied with in P). Assume that M is a weakly stable matching for P but M is not a super-stable matching for P'. Then there must exist married couples $(m,w'),(m',w) \in M$ such that in P', one of the following hold:

- m strictly prefers w to w' and w is indifferent between m and m'
  This is impossible, since by construction of P', even if w was indifferent between m and m' in P, she strictly prefers m' over m in P'.
- m strictly prefers w to w' and w strictly prefers m to m'
  This is impossible, since otherwise M would not be a weakly stable matching for P.
- w strictly prefers m to m' and m is indifferent between w and w'
  This is impossible, since by construction of P', even if m was indifferent between w and w' in P, he strictly prefers w' over w in P'.
- w strictly prefers m to m' and m strictly prefers w to w'
  This is impossible, since otherwise M would not be a weakly stable matching for P.
- m is indifferent between w and w' and w is indifferent between m and m'
  This is impossible, again by construction of P'.

Since all cases are impossible, we've reached a contradiction, and conclude that M must be a super-stable matching for P'.

4. Karl and Emma fight!

(a) Karl and Emma are having a disagreement regarding the traditional propose-and-reject algorithm. They both agree that it favors men over women. But they disagree about what, if anything, can be done without changing the ritual form of men proposing, women rejecting, and people getting married when there are no more rejections.

Karl mansplains: "It's hopeless. Men are obviously going to propose in the order of their preferences. It's male optimal so why would they do anything else? As far as the women are concerned, given that they face a specific choice of proposals at any given time, they are obviously going to select the suitor they like the most. So unless we smash the system entirely, it is going to keep all women down."

Emma says: "People are more perceptive and forward-looking that you think. Women talk to each other and know each other's preferences regarding men. They can also figure out the preferences of the men they might be interested in. A smart and confident woman should be able to do better for herself in the long run by not trying to cling to the best man she can get at the moment. By rejecting more strategically, she can simultaneously help out both herself and her friends."

Is Emma ever right? If it is impossible, prove it. If it is possible, construct and analyze an example (a complete set of people and their preference lists) in which a particular woman acting on her own (by not following the ordering of her preference list when deciding whether to accept or reject among multiple proposals) can get a better match for herself without hurting any other woman. Show how she can do so. The resulting pairing should also be stable.

Solutions: Emma is right. Here is an example of six people, three men $(1,2,3)$ and three women (A,B,C), together with their respective preference lists.

Man	Preference List
1	$A > C > B$
2	$A > B > C$
3	$C > A > B$

Woman	Preference List
A	$3 > 1 > 2$
B	$2 > 3 > 1$
C	$1 > 2 > 3$

We know what happens when we run the traditional propose-and-reject algorithm with these preference lists. We get the pairing $\{(1,A),(2,B),(3,C)\}$. But here, we can see that both A and C can do better.

Now, woman A looks at the preference lists of all the men and women and notices something. If she doesn't cling to the best she can get at the moment, she can end up with someone better! Further, she can do so without hurting B and C. Let's see how this plays out.

On day 1 of the traditional algorithm, we have the following proposals. The only change here is that A now rejects 1 instead of 2 even though she likes 1 more among them.

Day	1
A	1, (2)
B	
C	(3)

Now, we continue from as normal, using the traditional propose-and-reject algorithm. Here are the remaining days.

Day	1	2	3	4
A	1, (2)	(2)	2, (3)	(3)
B				(2)
C	(3)	(1), 3	(1)	(1)

The pairing produced from the run above is $\{(3,A),(2,B),(1,C)\}$. We see that woman A acting on her own on day one changed the face of the game for the women.

Is this pairing stable? Well, of course it is! Each woman ends up with the man she likes the most.

In conclusion, we can say with conviction that Emma was indeed right! A forward-thinking woman can potentially improve the ostensibly bleak outcome of the traditional propose-and-reject algorithm by strategically rejecting in the early stages.

(b) Karl and Emma have another disagreement! Karl claims that if a central authority was running the propose-and-reject algorithm then cheating the system might improve the cheater's chances of getting the more desirable candidate. The cheater need not care about what happens to the others.
Karl says: "Lets say there exists a true preference list. A prefers 1 to 2 but both are low on her preference list. By switching the reported preference order among 1 and 2, she can end up with 3 whom she prefers over 1 and 2 which wasn't possible if she did not lie. Isn't that cool?"
Emma responds: "That's impossible! In the traditional propose-and-reject algorithm switching the preference order 1 and 2 cannot improve A's chance to end up with 3."
Either prove that Emma is right or give an example of set of preference list for which a switch would improve A's husband (that is, she gets matched with 3), and hence proving Karl is right.

Solutions: Assume we have three men 1, 2 and 3 and three women A, B, and C with preferences as given in the table below. Row A shows true preferences of A, while in row A' she pretends she prefers 2 over 1.

Man	Preference List
1	$A > C > B$
2	$A > B > C$
3	$C > A > B$

Woman	Preference List
A	$3 > 1 > 2$
A'	$3 > 2 > 1$
B	$1 > 3 > 2$
C	$1 > 3 > 2$

First let us consider one possible execution of the traditional propose-and-reject algorithm with true preference list of A. Then we get the following pairing: $(1,A)$, $(2,B)$ and $(3,C)$.

Woman	Day 1	Day 2
A	(1), 2	(1)
B		(2)
C	(3)	(3)

If the same algorithm is run with the false preference list A' we get $(1,C)$, $(2,B)$ and $(3,A)$.

Woman	Day 1	Day 2	Day 3	Day 4
A	1,(2)	(2)	2, (3)	(3)
B				(2)
C	(3)	3, (1)	(1)	(1)

This way A got the man she wanted most (3) and also helped C get man 1, who is highest on C's preference list without actually intending to help.

5. TeleBears

In the Course Enrollment Problem, we are given n students and m discussion sections. Each discussion section u has some number, q_u of seats, and we assume that the total number of students is larger than the total number of seats (i.e. $\sum_{u=1}^{m} q_u < n$). Each student ranks the m discussion sections in order of preference, and the instructor for each discussion ranks the n students. Our goal is to find an assignment of students to seats (one student per seat) that is *stable* in the following sense:

- There is no student-section pair (s,u) such that s prefers u to her allocated discussion section and the instructor for u prefers s to one of the students assigned to u. (This is like the stability criterion for Stable Marriage: it says there is no student-section pair that would like to change the assignment.)

- There is no discussion section u for which the instructor prefers some unassigned student s to one of the students assigned to u. (This extends the stability criterion to take account of the fact that some students are not assigned to discussions.)

Note that this problem is almost the same as the Stable Marriage Problem, with two differences: (i) there are more students than seats; and (ii) each discussion section generally has more than one seat.

(a) Explain how to modify the propose-and-reject algorithm so that it finds a stable assignment of students to seats.

Solutions: We will extend the propose-and-reject algorithm given in the lecture notes. Students will play the role of men and discussion section instructors will play the role of women. Instead of keeping a single person as in the original algorithm, each discussion section instructor will keep a *waitlist* of size equal to its quota.

Note that there are other valid ways to modify the propose-and-reject algorithm such that a stable assignment is produced. One way is to have the instructors playing the role of men and the students playing the role of women, with the difference that now we have men proposing to up to q_u women each day. It is also possible to "expand" each instructor by the size of his or her quota by making q_u copies of instructor u and adding empty discussions for the unassigned students.

The extended procedure for students proposing and instructors keeping a waitlist works as follows:

- All students apply to their first-choice discussion section.
- Each discussion section u with a quota of q_u, then places on its waitlist the q_u applicants who rank highest (or all the applicants if there are fewer than q_u of them) and rejects all the rest.
- Rejected applicants then apply to their second-choice discussion section, and again each discussion section instructor u selects the top q_u students from among the new applicants AND those on its waitlist; it puts the selected students on its new waitlist, and rejects the rest of its applicants (including those who were previously on its waitlist but now are not).
- The above procedure is repeated until every applicant is either on a waitlist or has been rejected from every discussion section. At this point, each discussion section admits everyone on its waitlist.

(b) State a version of the Improvement Lemma (see Lecture Note 4) that applies to your algorithm, and prove that it holds.

Solutions:

Improvement Lemma Assume that discussion sections maintain their waitlist in decreasing order of their preference for the students on the lists. Let $\oplus$ be the "null" element, which we will use as a placeholder in waitlist positions not yet assigned to a student. For any discussion section u, let q_u be its quota and let $s_i^k \in \{Students\} \cup \{\oplus\}$ be the student in the i'th position on the waitlist after the k'th round of the algorithm. (If there are fewer than q_u students on the list, we fill the bottom of the list with $\oplus$'s.) Then for all i and k, the discussion section instructor likes s_i^{k+1} at least as much as s_i^k. (Here we assume that the discussion section instructor prefers any student to no student.)

In short, the lemma says that no position in the list ever gets worse for the discussion section instructor as the algorithm proceeds. As in the Lecture Notes, we use the Well-Ordering Principle and prove the lemma by contradiction:

Suppose that the jth day, where $j > k$, is the first counterexample where, for index $i \leq q_u$, discussion section u, has either nobody or some student $\hat{s}$ inferior to s_i^k. Then on day $j-1$, the instructor has some

student $\tilde{s}$ on a string that they like at least as much as s_i^k. Following the algorithm, $\tilde{s}$ still "proposes" to the instructor of section u on day j since they said "maybe" the previous day. Therefore, the instructor has the choice of at least one student on the jth day, and his or her best option is at least as good as $\tilde{s}$, so the instructor would have chosen $\tilde{s}$ over $\hat{s}$. Therefore on day j, the instructor *does* have a student that they like at least as much as s_i^k in waitlist position $i \leq q_u$. This contradicts our initial assumption.

(c) Use your Improvement Lemma to give a proof that your algorithm terminates, that every seat is filled, and that the assignment your algorithm returns is stable.

Solutions: First, the algorithm terminates. This follows by similar reasoning to the original propose-and-reject algorithm: in each round (except the last), at least one discussion section is crossed off the list of some rejected students.

Second, every seat is filled. Suppose some discussion section u has an unfilled seat at the end. Then the total number of students who applied to u must have been fewer than its quota q_u (since the Improvement Lemma in part(b) ensures that a waitlist slot, once filled, will never later be unfilled). But the only students who do *not* apply to u are those who find a slot in some other discussion section. And since we are told that there are more students than seats, the number of students applying to u must be at least q_u.

Finally, the assignment is stable:

- Suppose there is a student-section pair (s,u) such that s prefers u to her discussion section u' in the final allocation. Then s must have proposed to u prior to proposing to u', and was rejected by u. Thus immediately after rejecting s, u must have had a full waitlist in which every student was preferred to s. By the Improvement Lemma, the same holds at all future times and hence at the end. Thus u does not prefer s to any of its assigned students, as required.
- Suppose s is a student left unassigned at the end. Consider any discussion section u. Since s must have applied to and have been rejected from all discussion sections, this holds in particular for u. Reasoning similar as in the previous case, using the Improvement Lemma, we see that in the final allocation, u prefers all its students to s. Therefore u does not prefer s to any of its assigned students, as required.

This concludes the proof that our algorithm finds a stable assignment when terminates.

6. Long Courtship

(a) Run the traditional propose-and-reject algorithm on the following example:

Man	Preference List
1	$A > B > C > D$
2	$B > C > A > D$
3	$C > A > B > D$
4	$A > B > C > D$

Woman	Preference List
A	$2 > 3 > 4 > 1$
B	$3 > 4 > 1 > 2$
C	$4 > 1 > 2 > 3$
D	$1 > 2 > 3 > 4$

Solutions: The stable pairing reached by male propose-and-reject algorithm is $\{(1,D),(2,A),(3,B),(4,C)\}$.

Woman	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
A	1,(4)	(4)	(4)	4,(3)	(3)	(3)	3,(2)	(2)	(2)	(2)
B	(2)	2,(1)	(1)	(1)	1,(4)	(4)	(4)	4,(3)	(3)	(3)
C	(3)	(3)	3,(2)	(2)	(2)	2,(1)	(1)	(1)	1,(4)	(4)
D										(1)

(b) We know from the notes that the propose-and-reject algorithm must terminate after at most n^2 proposals. Prove a sharper bound showing that the algorithm must terminate after at most $n(n-1)+1$ proposals. Is this instance a worst-case instance for $n=4$? How many days does the algorithm take on this instance?

Solutions:

If we consider a scenario where every man has proposed to $n-1$ women, then either every woman has received a proposal and thus every woman has accepted a proposal and the algorithm terminates, or there is one woman who has never received a proposal. Since there are n men, the maximum number of possible proposals must be $n(n-1)+1$, where 1 is added to account for the last woman to whom a proposal might be extended on the last day.

In the example above there were $13 = 4(4-1)+1$ proposals in 10 days.

7. Better Off Alone

In the stable marriage problem, suppose that some men and women have standards and would not just settle for anyone. In other words, in addition to the preference orderings they have, they prefer being alone to being with some of the lower-ranked individuals (in their own preference list). A pairing could ultimately have to be partial, i.e., some individuals would remain single.

The notion of stability here should be adjusted a little bit. A pairing is stable if

- there is no paired individual who prefers being single over being with his/her current partner,
- there is no paired man and paired woman that would both prefer to be with each other over their current partners, and
- there is no single man and single woman that would both prefer to be with each other over being single.

(a) Prove that a stable pairing still exists in the case where we allow single individuals. You can approach this by introducing imaginary mates that people "marry" if they are single. How should you adjust the preference lists of people, including those of the newly introduced imaginary ones for this to work?

Solutions: Following the hint, we introduce an imaginary mate (let's call it a robot) for each person. Note that we introduce one robot for each individual person, i.e. there are as many robots as there are people. For simplicity let us say each robot is owned by the person we introduce it for.

Each robot is in love with its owner, i.e. it puts its owner at the top of its preference list. The rest of its preference list can be arbitrary. The owner of a robot puts it in his/her preference list exactly after the last person he/she is willing to marry. i.e. owners like their robots more than people they are not willing to marry, but less than people they like to marry. The ordering of people who someone does not like to marry as well as robots he/she does not own is irrelevant as long as they all come after their robot.

To illustrate, consider this simple example: there are three men $1,2,3$ and three women A,B,C. The preference lists for men is given below:

Man	Preference List
1	$A > B$
2	$B > A > C$
3	C

and the following depicts the preference lists for women:

Woman	Preference List
A	1
B	$3 > 2 > 1$
C	$2 > 3 > 1$

In this example, 1 is willing to marry A and B and he likes A better than B, but he'd rather be single than to be with C. On the other side B has a low standard and does not like being single at all. She likes 3 first, then 2, then 1 and if there is no option left she is willing to be forced into singleness. On the other hand, A has pretty high standards. She either marries 1 or remains single.

According to our explanation we should introduce a robot for each person. Let's name the robot owned by person X as R_X. So we introduce male robots R_A, R_B, R_C and female robots R_1, R_2, R_3. Now we should modify the existing preference lists and also introduce the preference lists for robots.

According to our method, 1's preference list should begin with his original preference list, i.e. $A > B$. Then comes the robot owned by 1, i.e. R_1. The rest of the ordering, which should include C and R_2, R_3 does not matter, and can be arbitrary.

For B, the preference list should begin with $3 > 2 > 1$ and continue with R_B, but the ordering between the remaining robots (R_A and R_C) does not matter.

What about robots' preference lists? They should begin with their owners and the rest does not matter. So for example R_A's list should begin with A, but the rest of the humans/robots (B, C, R_1, R_2, and R_3) can come in any arbitrary order.

So the following is a list of preference lists that adhere to our method. There are arbitrary choices which are shown in bold (everything in bold can be reordered within the bold elements).

Man	Preference List
1	$A > B > R_1 > \mathbf{3} > \mathbf{R_3} > \mathbf{R_2}$
2	$B > A > C > R_2 > \mathbf{R_1} > \mathbf{R_3}$
3	$C > R_3 > \mathbf{R_1} > \mathbf{R_3} > \mathbf{A} > \mathbf{B}$
R_A	$A > \mathbf{B} > \mathbf{C} > \mathbf{R_1} > \mathbf{R_2} > \mathbf{R_3}$
R_B	$B > \mathbf{R_1} > \mathbf{R_2} > \mathbf{R_3} > \mathbf{A} > \mathbf{C}$
R_C	$C > \mathbf{A} > \mathbf{R_2} > \mathbf{B} > \mathbf{R_1} > \mathbf{R_3}$

and the following depicts the preference lists for women and female robots:

Woman	Preference List
A	$1 > R_A > \mathbf{3} > \mathbf{R_B} > \mathbf{2} > \mathbf{R_C}$
B	$3 > 2 > 1 > R_B > \mathbf{R_C} > \mathbf{R_A}$
C	$2 > 3 > 1 > R_C > \mathbf{R_A} > \mathbf{R_B}$
R_1	$1 > \mathbf{R_B} > \mathbf{2} > \mathbf{R_C} > \mathbf{3} > \mathbf{R_A}$
R_2	$2 > \mathbf{R_A} > \mathbf{R_C} > \mathbf{1} > \mathbf{3} > \mathbf{R_B}$
R_3	$3 > \mathbf{2} > \mathbf{1} > \mathbf{R_A} > \mathbf{R_C} > \mathbf{R_B}$

Now let us prove that a stable pairing between robots and owners actually corresponds to a stable pairing (with singleness as an option). This will finish the proof, since we know that in the robots and owners case, the propose and reject algorithm will give us a stable matching.

It is obvious that to extract a pairing without robots, we should simply remove all pairs in which there is at least one robot (two robots can marry each other, yes). Then each human who is not matched is declared to be single. It remains to check that this is a stable matching (in the new, modified sense). Before we do that, notice that a person will never be matched with another person's robot, because

if that were so he/she and his/her robot would form a rogue couple (the robot's love is there, and the owner actually likes his/her robot more than other robots).

i. No one who is paired would rather break out of his/her pairing and be single. This is because if that were so, that person along with its robot would have formed a rogue couple in the original pairing. Remember, the robot loves its owner more than anything, so if the owner likes it more than his/her mate too, they would be a rogue couple.

ii. There is no rogue couple. If a rogue couple m and w existed, they would also be a rogue couple in the pairing which includes robots. If neither m nor w is single, this is fairly obvious. If one or both of them are single, they prefer the other person over being single, which in the robots scenario means they prefer being with each other over being with their robot(s) which is their actual match.

This shows that each stable pairing in the robots and humans setup gives us a stable pairing in the humans-only setup. It is noteworthy that the reverse direction also works. If there is a stable pairing in the humans-only setup, one can extend it to a pairing for robots and humans setup by first creating pairs of owners who are single and their robots, and then finding an arbitrary stable matching between the unmatched robots (i.e. we exclude everything other than the unmatched robots and find a stable pairing between them). To show why this works, we have to refute the possibility of a rogue pair. There are three cases:

i. A human-human rogue pair. This would also be rogue pair in the humans-only setup. The humans prefer each other over their current matches. If their matches are robots, that translates to them preferring each other over being single in the humans-only setup.

ii. A human-robot rogue pair. If the human is matched to his/her robot, our pair won't be a rogue pair since a human likes his/her robot more than any other robot. On the other hand if the human is matched to another human, he/she prefers being with that human over being single which places that human higher than any robot. Again this refutes the human-robot pair being rogue.

iii. A robot-robot rogue pair. If both robots are matched to other robots, then by our construction, this won't be a rogue couple (we explicitly selected a stable matching between left-alone robots). On the other hand, if either robot is matched to a human, that human is its owner, and obviously a robot loves its owner more than anything, including other robots. So again this cannot be a rogue pair.

This completes the proof.

(b) As you saw in the lecture, we may have different stable pairings. But interestingly, if a person remains single in one stable pairing, s/he must remain single in any other stable pairing as well (there really is no hope for some people!). Prove this fact by contradiction.

Solutions: We will perform proof by contradiction. Assume that there exists some man m_1 who is paired with a woman w_1 in stable pairing S and unpaired in stable pairing T. Since S is a stable pairing and m_1 is unpaired, w_1 must be paired in T with a man m_2 whom she prefers over m_1. (If w_1 were unpaired or paired with a man she does not prefer over m_1, then (m_1, w_1) would be a rouge couple, which is a contradiction.)

Since m_2 is paired with w_1 in T, he must be paired in S with some woman w_2 whom m_2 prefers over w_1. This process continues (w_2 must be paired with some m_3 in T, m_3 must be paired with some w_3 in S, etc.) until all persons are paired. Since this requires m_1 to be paired in T, where he is known to be unpaired, we have reached a contradiction. Therefore, our assumption must be false, and there cannot exist some man who is paired in a stable pairing S and unpaired in a stable pairing T. A similar argument can be used for women.

Since no man or woman can be paired in one stable pairing and unpaired in another, every man or woman must be either paired in all stable pairings or unpaired in all stable pairings.

Here is another possible proof:

We know that some male-optimal stable pairing exists. Call this pairing M. We first establish two lemmas.

Lemma 1. If a man is single in male-optimal pairing M, then he is single in all other stabling pairings.

Proof. Assume there exists a man that is single in M but not single in some other stable pairing M'. Then M would not be a male-optimal pairing, so this is a contradiction.

Lemma 2. If a woman is paired in male-optimal pairing M, she is paired in all other stable pairings.

Proof. Assume there exists a woman that is paired in M but single in some other stable pairing M'. Then M would not be female-pessimal, so this is a contradiction.

Let there be k single men in M. Let M' be some other stable pairing. Then by Lemma 1, we know single men in M' will be greater than or equal to k. We also know that there are $n-k$ paired men and women in M. Then by Lemma 2, we know that the number of paired women in M' will be greater than or equal to $n-k$.

Now, we want to prove that if a man is paired in M, then he is paired in every other stable pairing. We prove this by contradiction. Assume that there exists a man m that is paired in M but is single in some other stable pairing M'. Then there must be strictly greater than k single men in M', and thus strictly greater than k single women in M'. Since there are strictly greater than k single women in M', there must be strictly less than $n-k$ paired women in M'. But this contradicts that the number of paired women in M' will be greater than or equal to $n-k$.

We also have to prove that if a woman is single in M, then she must be single every other stable pairing. We again prove this by contradiction. Assume that there exists a woman w that is single in M and paired in some other stable pairing M'. Then there are strictly greater than $n-k$ paired women in M', which means there are strictly greater than $n-k$ paired men in M'. This means there must be strictly less than k single men in M'. But this contradicts that the number of single men in M' will be greater than or equal to k.

Since we have proved both 1) If a man is single in M then he is single in every other stable pairing and 2) If a man is paired in M then he is paired in every other stable pairing (note that the contrapositive of this is if a man is single in any other stable pairing, then this man is single in M), we know that a man is single in M if and only if he is single in every other stable pairing. Similarly, since we have proved both 1) If a woman is single in M then she is single in every other stable pairing and 2) If a woman is paired in M then she is paired in every other stable pairing, we know that a woman is single in M if and only if she is single in every stable pairing. Thus we have proved that if a person is single in one stable pairing, s/he is single in every stable pairing.

8. Write Your Own Problem

Write your own problem related to this week's material and solve it. You may still work in groups to brainstorm problems, but each student should submit a unique problem. What is the problem? How to formulate it? How to solve it? What is the solution?