

CS375 Lecture 5 (Sep 28)

📅 Date	September 28, 2026
--------	--------------------

Scribe notes for @September 28, 2026, by Francis O'Hara Aidoo. Typed from my handwritten notes.

Administrative Announcements

Midterm 1 Details

Topics that will be important for the midterm:

- Exchange argument
 - There will definitely be an exchange argument on the exam.
 - Memorizing is not the point. Know what an exchange argument is, and do enough of them that it feels comfortable.
- Use induction to show an algorithm is correct
 - Guaranteed to be on the exam. It will be a simple loop-based iterative algorithm, like how we did insertion sort.
- Fundamentals of runtime
- Today's topic (stable matching) will also be on the midterm in some capacity.

Location: Miller 008 (waiting on confirmation, will be posted on Classroom).

Other announcements

- An old homework assignment has been updated with more practice problems of the kind "show this iterative algorithm works". Doing them is not required. It will be posted on Classroom.
- Wednesday (Sep 30) is a review session. Email Prof. Bender before class with what you would like covered. The more specific the better, for example a particular homework problem.

- No graded quiz on Wednesday. Instead there will be an ungraded practice problem that looks like something that could be on the midterm.
-

Deferred Acceptance (Stable Marriage)

- The last lecture on greedy algorithms.
- One of Prof. Bender's favorite algorithms, because it controls a lot of things that happen in the world today:
 - Used to decide where medical students applying for residency positions end up.
 - Was used to assign CS students to classes two years ago. At the start of Workday, the department had everyone fill out their preferences for the classes they wanted, then ran this algorithm behind the scenes to decide who got which class.

The problem

Uses the language of the original paper (not representative of the world today).

Assumptions:

- There is an equal number of men and women.
- Everyone is heterosexual, only for the purposes of this problem. Otherwise the math gets much harder. The homework looks at the more general case.
- Every man has a preference ranking over every woman, and every woman has a preference ranking over every man.
- It is only a ranking. Nobody says how much more or less they like someone.

Goal: match everyone up in a way that is fair in some reasonable sense.

Input: n men, n women.

For each man/woman, a preference ranking over all women/men.

Output: A stable matching over all the men and women.

What "unstable" means: a solution is unstable if there is a man and a woman who prefer each other over their assigned partners.

Such a pair has a reason to leave their assigned partners for each other. The paper calls them a "rogue couple".

For now we only care about stability, not optimality.

Toy example

$$M = \{A, B, C\} \quad W = \{X, Y, Z\}$$

X	X	Y	:	C	A	A
AY	BY	CX	:	XA	YC	ZB
Z	Z	Z	:	B	B	C

Meaning

Each column is one person's preference list, read from top (most preferred) to bottom (least preferred). The men (A, B, C) are on the left of the dotted line and the women (X, Y, Z) are on the right. For example, A prefers X, then Y, then Z, and X prefers C, then A, then B. The same lists as a table:

Person	1st	2nd	3rd
A	X	Y	Z
B	X	Y	Z
C	Y	X	Z
X	C	A	B
Y	A	C	B
Z	A	B	C

<u>A-X</u>	B-Y	<u>C-Z</u>
Next round:		
A-Z	<u>B-Y</u>	<u>C-X</u>
<u>A-Z</u>	<u>B-X</u>	C-Y
A-X	B-Z	C-Y

Meaning

Start from an arbitrary matching (A is paired with X, B is paired with Y, C is paired with Z) and try to fix each instability with a swap:

1. A-X, B-Y, C-Z. C and X prefer each other over their partners (C prefers X to Z, and X prefers C to A). Swap them together.
 2. A-Z, B-Y, C-X. A went from very happy to very unhappy. C and Y prefer each other (C prefers Y to X, and Y prefers C to B). Swap.
 3. A-Z, B-X, C-Y. A and X prefer each other (A prefers X to Z, and X prefers A to B). Swap.
 4. A-X, B-Z, C-Y. No instabilities left, so this matching is stable.
- Can we always start from a random pairing like above and keep moving until we eventually arrive at a stable matching?
 - No. You can get stuck in a cycle and end up back where you started (this shows up on the homework).
 - We want an algorithm that is guaranteed to reach a stable matching in a finite number of rounds.
 - Is there always a stable matching?
 - Yes, there is always at least one.
 - Having two separate groups matters. In the one-sided version, where every person ranks every other person, a stable solution is not guaranteed.

Gale-Shapley Algorithm (Deferred Acceptance)

The algorithm runs in rounds and keeps building up a partial solution until everyone is matched.

Until everyone is matched:

Have every man who does not yet have a partner 'propose' to the woman he values the most among those he has not yet proposed to.

Every woman tentatively accepts the proposal she values the most (possibly rejecting a man she tentatively accepted before).

Observations:

- Over time, each woman's partner only gets better in her eyes.
- Over time, each man moves down his list. He can be tentatively accepted and then rejected later.
- Who does this algorithm favor? Most of the class guessed the women (about 95% guess that every semester).

Does the algorithm terminate? Yes.

- Several men can propose to the same woman in one round (for example A and B both propose to X on the first night). She keeps the one she prefers.
- Once a woman tentatively accepts someone, she always has a partner from then on.
- Suppose a man ends up with no partner. Then some woman has no partner either. He proposed to her at some point, and she would only have rejected him for someone she preferred, so she would have a partner. Contradiction.

Conclusion: everyone ends up matched, so the algorithm terminates.

Is the result stable? Yes.

We will show this with a proof by contradiction (same flavor as an exchange argument).

If Deferred Acceptance (D.A.) isn't stable, there is a pair (m, w) who prefer each other over their assigned partners. Let w' be m 's partner: m prefers w over w' .

Men go down their lists, so m proposed to w before w' . At some point, w rejected him. At that moment, w rejected him for someone she preferred more, and she continued to have even better partners afterwards. So w prefers whoever she ends up with over m , and (m, w) is not an instability.

Conclusion: there is no such pair, so the matching is stable.

Who does the algorithm favor?

The men. Every man gets his best possible partner, and every woman gets her worst.

Every single man ends up with their most preferred partner possible

What "possible" means: a partner is possible if there is some stable solution in which the two are matched.

So out of all stable solutions, every man gets the best partner he could have. It may not be his first choice, but it is his first choice that is possible.

Suppose otherwise: Look at the **first** man m rejected by a woman w for whom there is some stable solution S in which they are paired. Let m' be the man w rejected m for. Let w' be the woman m' is matched to in S .



Meaning

Left: w prefers m' over m . Right: m' prefers w over w' . The red circles show who each of them prefers. The green dashed circles show who each of them is matched with in S .

What do we know about their preferences?

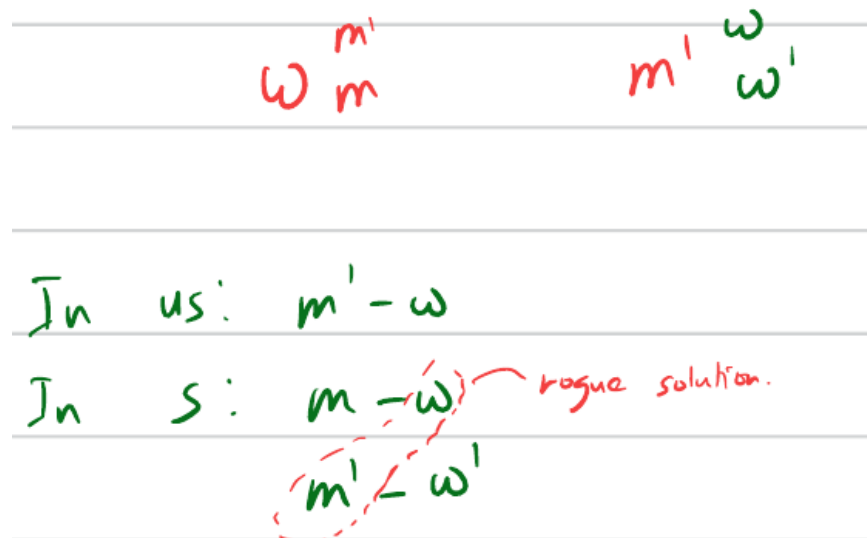
- w prefers m' over m , because she rejected m in favor of m' .
- m' prefers w over w' . If he preferred w' , he would have proposed to w' before w and been rejected by her. But w' is a possible partner for him (they are matched in S), so m' would have been the first man rejected by a possible partner, not m .
- S 's matching: $m - w$ and $m' - w'$.
- w and m' prefer each other over their partners in S , so S is not stable. No such S exists!

Conclusion: whenever a man is rejected by a woman in this algorithm, there is no stable solution in which they could have been paired. Every man ends up with his best possible partner.

Every woman ends up with their worst possible partner (out of all possible stable solutions)

Let m' be w 's partner in our solution. Let m be some man w prefers less than m' , and for whom there is a stable solution S in which they are matched.

Let w' be the partner of m' in this other solution S .



Meaning

Top: w prefers m' over m , and m' prefers w over w' . Bottom: the two matchings. The red dashed circle marks w and m' , who prefer each other over their partners in S .

- In ours: $m' - w$
- In S : $m - w$, $m' - w'$
- m' is matched with w in our solution, so w is his best possible partner (previous result). w' is also a possible partner for him, so m' prefers w over w' .
- w prefers m' over m , by assumption.
- So w and m' prefer each other over their partners in S . They are a rogue couple, so S is not stable.

Conclusion: S doesn't exist. No woman can be matched, in any stable solution, with a man she likes less than her partner here. Every woman ends up with her worst possible partner.

Remarks

- This is surprising, because the men's partners get worse over time and the women's get better. What the men are really doing is ruling out impossible partners until they reach the first one that works.
- The result is not about how far down their list someone ends up. It is about the best or worst partner among those possible in a stable solution.
- What if someone would rather have no partner than be with some of the people on their list? Put "no one" into their preference list at that point. The proof needs a little more work, but the same algorithm gives the same result.

Takeaway

- Deferred Acceptance terminates, produces a stable matching, and that matching is the best stable one for every man and the worst stable one for every woman.
- The proofs have the same flavor as an exchange argument: look at the first place another solution disagrees with ours, and show that the other solution

did something wrong. Here, the other solution turns out not to be feasible (it is not stable).