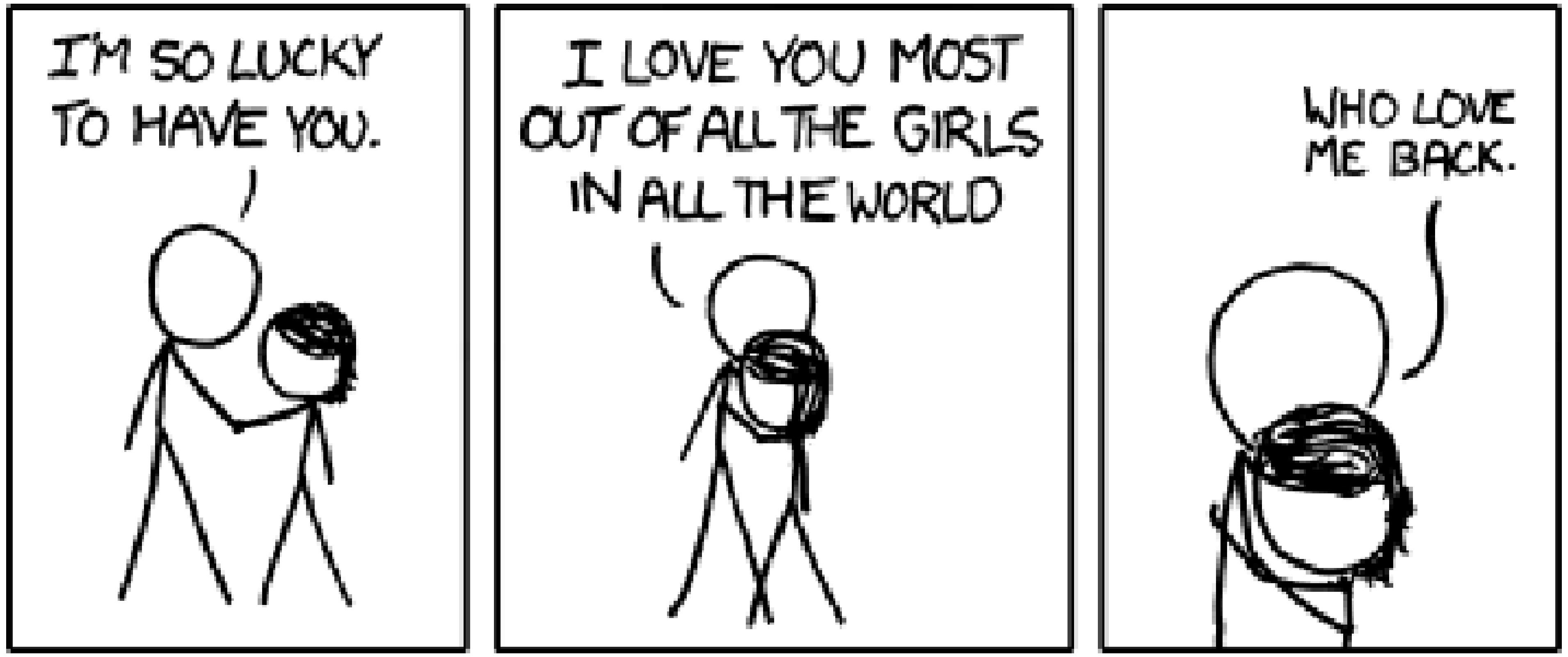
CSCI270 Discussion Week 1
Stable Matching

xkcd #770



Gale-Shapley Proposal Algorithm

- Maintain temporary assignments of "engagements"
- Start with the empty assignment
- Until there is no single man
    * Let m be any single man
    * Let w be the highest-ranked woman (according to his list) to whom he has never "proposed"
    * If w is single (unmatched) or prefers m over her current partner, then
        - match m and w (temporary engagement)
        - if w was previously matched, her old partner becomes single
    * otherwise, do nothing (but m will never propose to w again)
- finalize the temporary assignment

Complexity Analysis

- This algorithm terminates

Proof: Each proposal is made only (at most) once, because men go down the list. There are at most n^2 proposals total (each man for each woman). So there can be at most n^2 iterations.
If the algorithm didn't terminate, some man m would be single at the end.
Because we are assuming #men = #women, there would be a single woman w as well.
Because m exhausted all his proposals, he must have proposed to w. By the lemma, w can then not end up single.
==> Contradiction.
-- Lemma: Once a woman is proposed to, she will be partnered until the end. Her partners only improve only time (because she can reject ones she does not like better).

Complexity Analysis

1. Identify a free man
2. For a man m, identify the highest ranked woman to whom he has not yet proposed.
3. For a woman w, decide if w is engaged, and if so to whom
4. For a woman w and two men m & m', decide which man is preferred by w
5. Place a man back in the list of free men.

1. Identify a free man

Data structure	get (step1)	put (step 5)
Linked list	O(1)	O(1)
Queue	O(1)	O(1)
Stack	O(1)	O(1)
Array	O(1)	O(1)

2. For a man m, identify the highest ranked woman to whom he has not yet proposed

Keep an array Next[1...n] where Next[m] points to the position of the next woman that m will be proposing to on his pref list
Define men's preferences lists: ManPref[1...n,1...n], where ManPref[m,i] denotes the ith woman on man m's preference list
To find next woman w to whom m will be proposing to:
w = ManPref[m, Next[m]] // O(1)

1. Identify a free man $O(1)$
2. For a man m, identify the highest ranked woman to whom he has not yet proposed. $O(1)$
3. For a woman w, decide if w is engaged, and if so to whom $O(1)$
4. For a woman w and two men m & m', decide which man is preferred by w $O(1)$
5. Place a man back in the list of free men. $O(1)$

3. Determine woman w's status

Keep an array called current [1...n] where Current[w] is Null if w is single & set to m if w is engaged to m.
Takes O(1)

4. Determine which man is preferred by w

WomanPref_i:

3	8	4	37	1	..	...
1	2	3	4	5	...	

WomanRanking_i:

5		1	3	...
1	2	3	4	

Preparation before entering GS iterations

Create a Ranking array where Ranking [w,m] contains the rank of man m based on w's preferences

Preparation + GS iterations = O(n^2) + O(n^2)
Overall complexity = O(n^2)

Does it matter which man m proposes next in GS?
Answer happens to be no, and we will will characterize which matching is produced

Claim: Every execution of the GS algorithm (when men propose) results in the same stable matching regardless of the order in which men propose

Def: Woman w is a "valid partner" of a man m if there is a stable matching that contains the pair (m,w)

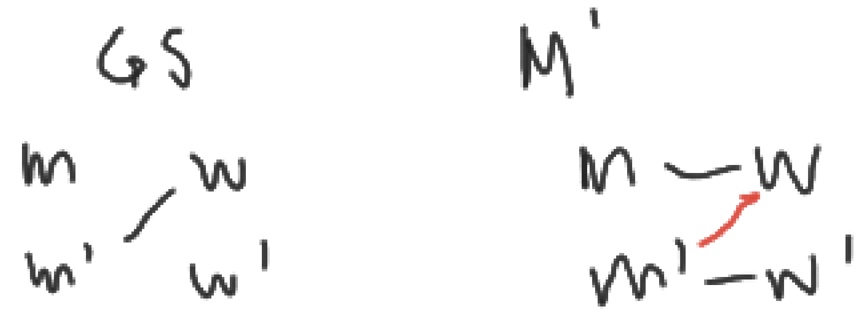
Def: m's "best valid partner" is the highest-ranked woman w in the set of valid partners for man m

Plan: to prove this claim, we will show that when men propose, they always end up with their best valid partner.

*Exercise: No two men have the same best valid partner.

Proof: by contradiction. Assume that some man m is not matched up with his best valid partner b(m).
Then, at some point, he must have been rejected/dumped by his b(m).
Look at the very first time that some man got rejected/dumped by a woman who was a valid partner. Let m be that man, and w the woman who rejected/dumped him.
Therefore, the man m' that w rejects/dumps m for is strictly higher on her list than m.
Because w was a valid choice for m, there exists some other stable matching M' in which m and w end up with each other.
Who is matched with m' in M'? Let's call her w'.

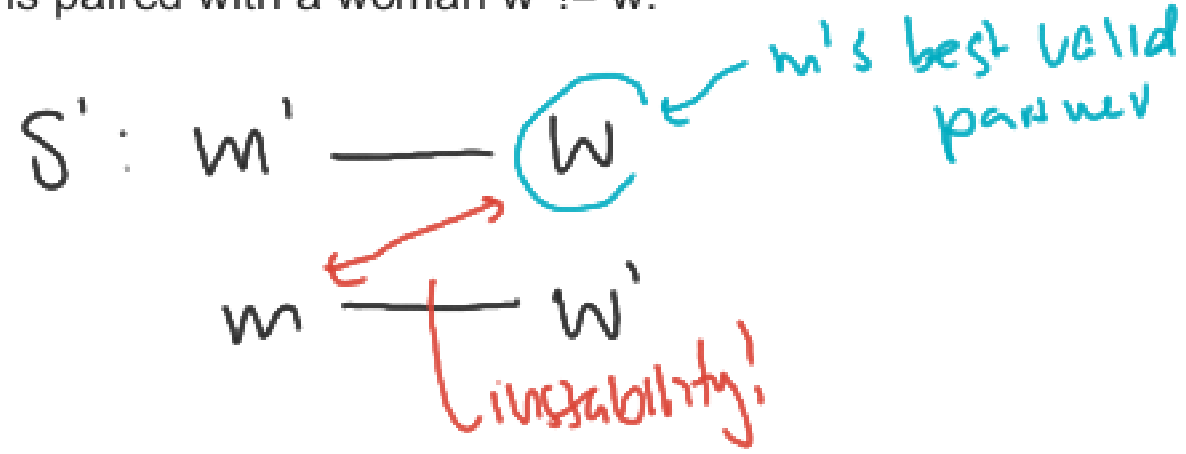
Who does m' prefer between w and w'?



Since we know that w prefers m' over m, if m' preferred w over w', M' would not be stable - (m', w) would prefer being with each other.
So we infer that m' actually prefers w' over w.
In GS, m' ended up paired with w.
So he must have proposed to w' earlier and been rejected/dumped.
This must have happened before w rejected/dumped m.
w' was a valid partner for m' (because they are partnered in M').
So we have a rejection of w' rejecting m' before w rejected m.
So w rejecting m was not the *first* rejection of some man by a woman in his set of valid partners.
==> Contradiction. QED

Claim: When men propose, women end up with their worst valid partners.

Proof: By contradiction. Suppose we end up with a matching where there is a pair (m,w) such that m is not the worst valid partner of w.
Then there is a some other stable matching S' in which w is paired with a man m' whom she likes less than m.
In S', m is paired with a woman w' != w:



Since w is the best valid partner of m, and w' is a valid partner of m, we see that m prefers w to w.
But from this, it follows that (m,w) is an instability in S', contradicting the claim that S' is stable, and hence contradicting our initial assumption.

Exercise 1) Four students, a, b, c, and d, are rooming in a dormitory. Each student ranks the others in strict order of preference. A "roommate matching" is defined as a partition of the students into two groups of two roommates each. A roommate matching is "stable" if no two students who are not roommates prefer each other over their roommate.
Does a stable roommate matching always exist? If yes, give a proof. Otherwise, give an example of roommate preferences where no stable roommate matching exists.

A stable matching need not exist.
Consider the following list of preferences. Note a, b, and c all prefer d the least.
- a: b > c > d
- b: c > a > d
- c: a > b > d

Now, there can only be 3 sets of disjoint roommate pairs.
- If the students are divided as (a,b) and (c,d), then (b,c) cause an instability, since c prefers b over d and b prefers c over a.
- If the students are divided as (a,c) and (b,d), then (a,b) cause an instability, since b prefers a over d and a prefers b over c.
- If the students are divided as (a,d) and (b,c), then (a,c) cause an instability, since c prefers a over b and a prefers b over c.

Thus every matching is unstable, and no stable matching exists with this list of preferences.

Exercise 2) Determine whether the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

For all n>=2, there exists a set of preferences for n men and n women such that in the stable matching returned by the GS algorithm when men are proposing, every man is matched with their most preferred woman.

True: Consider the case when every man has a different most preferred woman.

Exercise 3) Consider a stable marriage problem where the set of men is given by M = m1, m2, ..., mN, and the set of woman is W = w1, w2, ... wN. Consider their preference lists to have the following properties:

$$\forall w_i \in W : w_i \text{ prefers } m_i \text{ over } m_j \qquad \forall j > i$$

$$\forall m_i \in M : m_i \text{ prefers } w_i \text{ over } w_j \qquad \forall j > i$$

Prove that a unique stable matching exists for this problem.

We will prove that matching S where m_i is matched to w_i for all i=1,2,...,N is the unique stable matching in this case. We will use the notation S(a) = b to denote that in matching S, a is matched to b. It is evident that S is a matching because every man is paired with exactly one woman and vice versa. If any man m_j prefers w_k to w_j where k < j, then such a higher ranked woman w_k prefers her current partner to m_j. Thus, there are no instabilities and S is a stable matching.

Now let's prove that this stable matching is unique.
By way of contradiction, assume that another stable matching S' exists which is different from S. Therefore, there must exist some i for which S(w_i) = m_k, k != i. Let x be the minimum value of such an i.
Similarly, there must exist some j for which S(m_j) = w_l, j != l. Let y be the minimum value of such a j.
Since S'(w_i) = m_i for all i < x and S'(m_j) = w_j for all j < y, x=y.
S'(w_k) = m_k implies x < k. Similarly, S'(m_y) = w_l implies that y = x < l. Given the preference lists, m_y = m_x prefers w_x to w_l, and w_x prefers m_x to m_k. This is an instability. So S' cannot be a stable matching ==> Contradiction.