Lecture 5 Note Monday, 02/14/2022

Announcements:

- Happy Valentine's Day!
- The Lecture Recordings will be available on the following YouTube Playlists Link: https://youtube.com/playlist?list=PLZaTmV9UMKlgYpo2cAiMaEWxqyvbiXDFd

Stable Matching

References:
Algorithm Design - Chapter 1 section 1

Analysis of the G-S algorithm

- Does the program always terminate?
- Does it return a perfect matching?
- Is the matching stable?

Example of the G-S algorithm

Men's Preference					
	1st	2nd	3rd	4th	5th
m_1	w_2	w_3	w_4	w_5	w_1
m_2	w_1	w_4	w_5	w_2	w_3
m_3	w_1	w_5	w_4	w_3	w_2
m_4	w_2	w_1	w_3	w_4	w_5
m_5	w_4	w_3	w_5	w_1	w_2

Women's Preference					
	1st	2nd	3rd	4th	5th
w_1	m_5	m_3	m_2	m_1	m_4
w_2	m_2	m_1	m_4	m_5	m_3
w_3	m_1	m_5	m_2	m_3	m_4
w_4	m_4	m_1	m_3	m_2	m_5
w_5	m_3	m_4	m_1	m_2	m_5

- Run the C-S algorithm on this example:
  - Round 1: m_1 and m_4 propose to w_2, w_2 engaged with m_1 and rejected m_4.
    m_2 and m_3 propose to w_1, w_1 engaged with m_3 and rejected m_2.
    m_5 proposes to w_4, and w_4 engaged to m_5.
    You have pairs: $(m_1, w_2), (m_3, w_1), (m_5, w_4)$. m_2 and m_4 are rejected.
  - Round 2: m_2 proposes w_4, w_4 rejected m_5 and engaged with m_2.
    m_4 proposes w_1, w_1 rejected m_4 and remain engaged with m_4.
    You have pairs: $(m_1, w_2), (m_2, w_4), (m_3, w_1)$. m_4 and m_5 are rejected.
  - Round 3: m_4 and m_5 propose to w_3, w_3 engaged with m_5 and rejected m_4.
    You have pairs: $(m_1, w_2), (m_2, w_4), (m_3, w_1), (m_5, w_3)$. m_4 is rejected.
  - Round 4: m_4 proposes w_4, w_4 rejected m_2 and engaged with m_4.
    You have pairs: $(m_1, w_2), (m_3, w_1), (m_4, w_4), (m_5, w_3)$. m_2 is rejected.
  - Round 5: m_2 proposes to w_5, and w_5 engaged to m_2.
    You have pairs: $(m_1, w_2), (m_2, w_5), (m_3, w_1), (m_4, w_4), (m_5, w_3)$.
  - No more proposal can be made, the algorithm end.
  - Therefore, we got a "stable" matching:
    $(m_1, w_2), (m_2, w_5), (m_3, w_1), (m_4, w_4), (m_5, w_3)$.

- Review the analysis on what we did last class
  - A woman ends up engaged to the highest ranked man, highest ranked of all men who proposed to her.
  - During the running of the algorithm, the ranking of women in the preference list of the man who proposes gets worse.
  - [Theorem] For n men and n women, the G-S algorithm terminates after at most n^2 iterations in the loop.
  - [Lemma] If a man is free at some point in the execution of the algorithm, then there is a woman to whom he has not yet proposed.

Analysis of the G-S algorithm

- [Lemma] The G-S algorithm returns a perfect matching.
  Proof:
  - G-S algorithm will return a matching.
    - Since a woman is either free or engaged with only one man, so every woman would appear at most once in the pairing.
    - A man who engaged does not propose, so a man is either free or engaged to a woman, so every man would appear at most once in the pairing.
  - To show the matching is perfect. We can proof it by contradiction:
    - Assume the algorithm terminates with a free man (not perfect matching).
    - Then, by previous lemma, there is a woman whom that the free man has not yet propose to.
    - And we know that the algorithm terminates only if there is no more proposal, which means there is no free man who has not yet proposed to every woman.
    - This contradicts the assumption; we have a perfect matching.

- [Theorem] The result of Gale-Shapley algorithm is a stable matching.
  Proof by contradiction:
  - Assume that we have (m, w') and (m', w), $m, m' \in M, m \neq m'; w, w' \in W, w \neq w'$, in our matching, and there is an unstable pair (m, w).
    - which m and w prefer each other more than their current partners.
  - m prefer w to w', which implies m must propose to w before w'.
  - Then w was proposed by m, but didn't end up with him.
    - w will not end up with m for the following reasons:
      1) she was already engaged with someone better.
      2) she temporarily engaged to m, but rejected him later for someone better.
    - In both cases, w will end up with someone better than m, which will contradict the fact she prefer m to m' (her current partner).

Analysis of the G-S algorithm

- Does the G-S algorithm (men propose, women reject) favor men or women?
  - In G-S algorithm,
    - the partner for women can only get better.
    - the partner for men can only get worse.

- In the previous example, we did men propose on G-S algorithm.
  Let's find another stable matching by women propose.

Men's Preference					
	1st	2nd	3rd	4th	5th
m_1	w_2	w_3	w_4	w_5	w_1
m_2	w_1	w_4	w_5	w_2	w_3
m_3	w_1	w_5	w_4	w_3	w_2
m_4	w_2	w_1	w_3	w_4	w_5
m_5	w_4	w_3	w_5	w_1	w_2

Women's Preference					
	1st	2nd	3rd	4th	5th
w_1	m_5	m_3	m_2	m_1	m_4
w_2	m_2	m_1	m_4	m_5	m_3
w_3	m_1	m_5	m_2	m_3	m_4
w_4	m_4	m_1	m_3	m_2	m_5
w_5	m_3	m_4	m_1	m_2	m_5

  - Round1: w_1 proposes to m_5, w_2 proposes to m_2, w_3 proposes to m_1, w_4 proposes to m_4, and w_5 proposes to m_3. m_1 will engage with w_3, m_2 will engage with w_2, m_3 will engage with w_5, m_4 will engage with w_4, and m_5 will engage with w_1. No men get rejected; algorithm terminates.

- Compare the result with men propose.
  - For men propose, we got the stable matching: $(m_1, w_2), (m_2, w_5), (m_3, w_1), (m_4, w_4), (m_5, w_3)$.
  - For women propose, we got the stable matching: $(m_1, w_3), (m_2, w_2), (m_3, w_5), (m_4, w_4), (m_5, w_1)$.
  - If you circle the results on the preference tables, you will see that men get better choices in the men propose model. And women get better choices in the women propose model. In this particular example, women will all get their top choice in the women propose version.
- We found that the G-S algorithm actually favors the men (proposed side) then women (rejected side).

Analysis of the G-S algorithm

- Let's look from men's perspective.
  - For a man m,
    $valid(m) = \{set\ of\ women, w, that\ m\ is\ paired\ with\ w\ in\ some\ stable\ matings\}$, and
    $best(m) = the\ woman\ that\ most\ preferred\ by\ m\ in\ vaild(m)$.
- Prove that every man will match with his best valid partner.
  i.e., G-S algorithm returns a stable matching $S^* = \{(m_i, best(m_i)) \mid m_i \in M\}$.
  Proof by contradiction:
  - Assume there is a pair (m, w) in the matching S^* such that $w \neq best(m)$.
  - Since men propose in decreasing order, m must be rejected by $best(m)$, which either $best(m)$ engaged to a man m' better than m,
    or $best(m)$ was engaged to m, but rejected him later down the line for someone better, m'.
    In both cases, $best(m)$ prefer m' to m.
  - Since $best(m)$ is a valid partner for m, then there must exist a stable matching S' that $(m, best(m)) \in S'$.
  - Consider m' as the man engaged to $best(m)$ in S^*, since $best(m)$ is paired with m in S', m' must be paired with someone else, w', that $w' \neq best(m)$. m' must prefer $best(m)$ to w', since $(m', best(m)) \in S^*$.
  - In S', we have $(m, best(m))$ and (m', w'). But $(m', best(m))$ is an unstable matching, they prefer each other more than their current partners in S', which contradict that S' is a stable matching.

- Let's look from women's perspective.
  - For a woman w,
    $valid(w) = \{set\ of\ men, m, that\ w\ is\ paired\ with\ m\ in\ some\ stable\ matings\}$, and
    $worse(w) = the\ man\ that\ least\ preferred\ by\ w\ in\ vaild(w)$.
- Prove that every woman will match with her worse valid partner.
  i.e., G-S algorithm returns a stable matching $S^* = \{(worse(w_i), w_i) \mid w_i \in W\}$.
  Proof by contradiction:
  - Assume there is a pair (m, w) in the matching S^* such that $m \neq worse(w)$.
  - Then there is a stable matching S' that $(m', w) \in S', m' = worse(w)$,
    - i.e., w prefer m to m'.
  - In S', there exist a w' such that $(m, w') \in S', w' \neq w$.
  - By previous theorem, $w = best(m)$, since $(m, w) \in S^*$.
    - Which m prefer w to w'.
  - We have $(m', w) \in S'\ and\ (m, w') \in S'$, but (m, w) is an unstable matching, they prefer each other more than their current partners in S', which contradict that S' is a stable matching.

Problem Statement vs Algorithm

What's the difference between algorithm and problem statement?

- In algorithm class, many students mix the algorithm with the problem statement.
- Algorithm is the way to solve a problem, and there could be many ways to solve a problem. We should have the solution separate from the problem statement. Problem Statement are referring to the question. What's the input? What's the required output?
  How we approach and solve the problem is the algorithm.

- Stable matching is our first question. Hopefully this gives you an idea of what a problem statement looks like, what is the input, what is the output.
- And when you come up with an algorithm, you need to prove two things:
  1) prove that it's correct that our algorithm does the job, and
  2) prove that it does the job in a reasonable time, i.e., polynomial time. If it takes forever, then the algorithm is useless, even if it's correct.
- Other variants: Men-Women-Cats (3D matching).

Assignment/Project [5 %]

- Implement the G-S algorithm.

Men's Preference				
	1st	2nd	3rd	4th
m_1	w_2	w_3	w_4	w_1
m_2	w_1	w_4	w_2	w_3
m_3	w_1	w_3	w_4	w_2
m_4	w_2	w_1	w_3	w_4

Women's Preference				
	1st	2nd	3rd	4th
w_1	m_1	m_3	m_2	m_4
w_2	m_2	m_1	m_4	m_3
w_3	m_1	m_3	m_2	m_4
w_4	m_4	m_1	m_3	m_2

1) Run this example on your program, show evolution of each round.
2) Run this example again with the roles of men and women swapped (women propose, men reject), show evolution of each round.
3) Compare your result in part (1) and (2).

- **Submission instruction:**
  - You can implement the algorithm in any language you prefer.
  - I will suggest you put your code, comments, and answer to above questions on a Jupyter Notebook. However, it's not mandatory. You can have also sent the coding file with your implementation and answer the questions in email.
  - You will submit your work via email with the subject "**CS323 GS algorithm**". Please sent the email from your school email to xinying.chyn@qc.cuny.edu.
  - In the email, please include the following:
    a. explanation of how do your program import input.
    b. outputs of the about example for part (1) and (2)
    c. Your answer to part (3).
    d. the coding file with your implementation.
  - It's due Friday, February 18, 2022. No late submission.

What to expect or prepare for the next class:

- Graph

Reading Assignment

Algorithm Design: 1.1

Suggested Problems

- Algorithm Design – Chapter 1 – 1, 2, 8