TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

A Nobel Prize Algorithm: The Stable Marriage

Algorithms, Data and Security
A.Y. 2025/26

Valeria Cardellini

Global Governance, 3rd year
Science and Technology Major

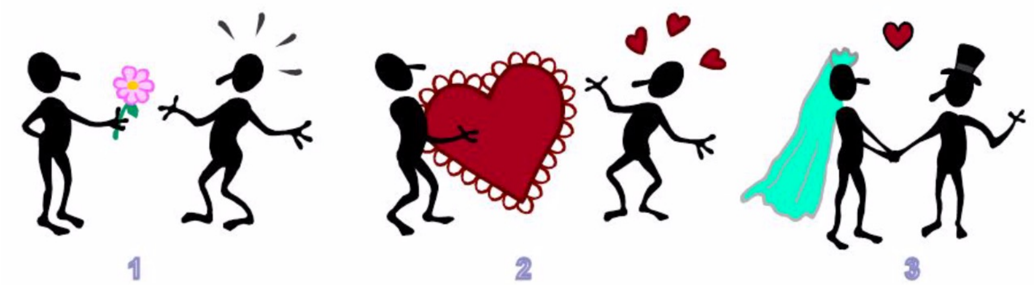
Matching problems

- Two-sided matching markets: matching occurs between two sets of agents based on preferences
- Matching professors and courses
- Matching rooms and courses
- Matching students and internships
- Matching users and servers
- And other examples when we aim to find a matching that is optimal from an individual perspective

A Nobel prize algorithm

- Lloyd Shapley (1923-2016): an American mathematician and Nobel Prize-winning economist
  - Contributed to mathematical economics and especially game theory
  - Won 2012 Nobel Memorial Prize in Economic Sciences with Alvin Roth “for the theory of stable allocations and the practice of market design”
  - One contribution is the **Gale-Shapley algorithm**

The stable marriage problem



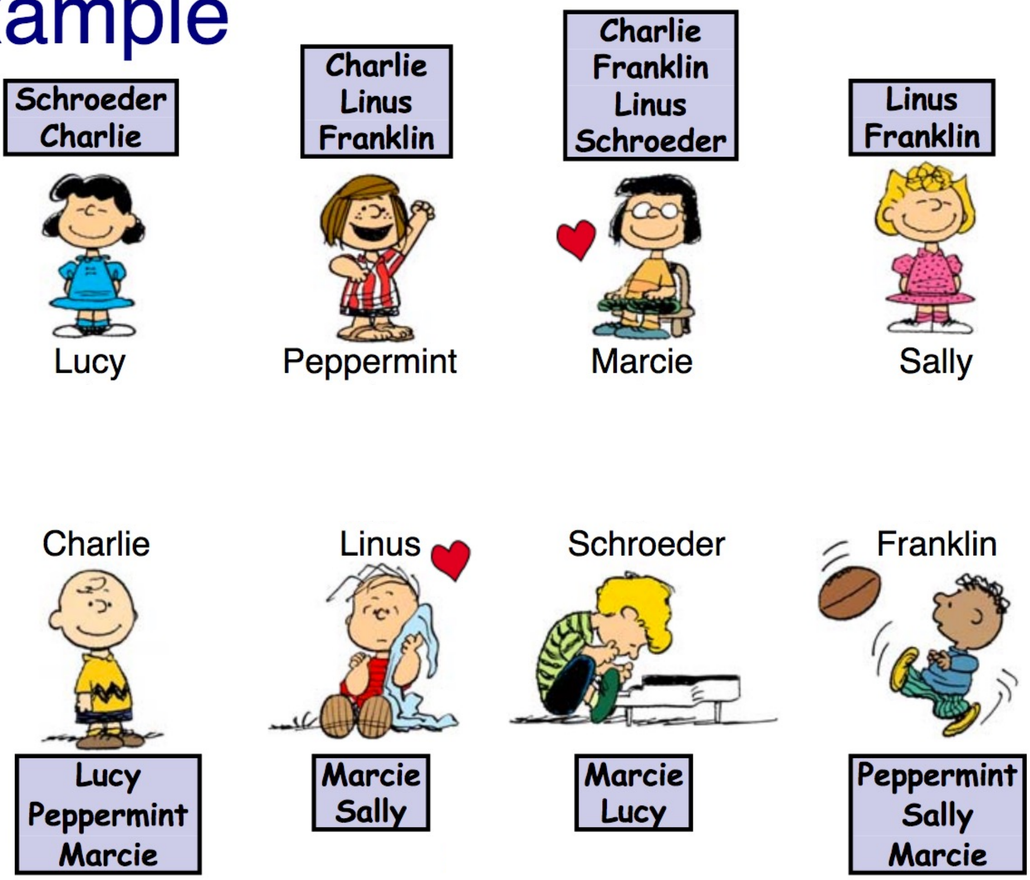
- Goal: match two sets of equal size (traditionally *n* men and *n* women, which are unmarried)
- Preferences: each individual ranks all members of the opposite set in order of preference
- Also known as stable matching problem

The stable marriage problem

- Does there exist and can we find a stable matching (stable opposite-sex marriage)?
  - That is, a matching of men and women, such that there is no pair of a man and a woman who both prefer each other to their current partner in the matching

Note: we will not go over original description, which was not gender neutral

Example



Blocking pairs and stable marriage

- Matching *M*: a set of pairs where each person has exactly one partner
- Given a matching *M* of men and women, a pair (*m*, *w*), where *m* is a man and *w* is a woman, is a **blocking pair** if and only if
  - *m* and *w* are not partners in *M*
  - *m* prefers *w* over his current partner and *w* prefers *m* over her current partner
- A matching *M* is **stable** if and only if no blocking pairs exist

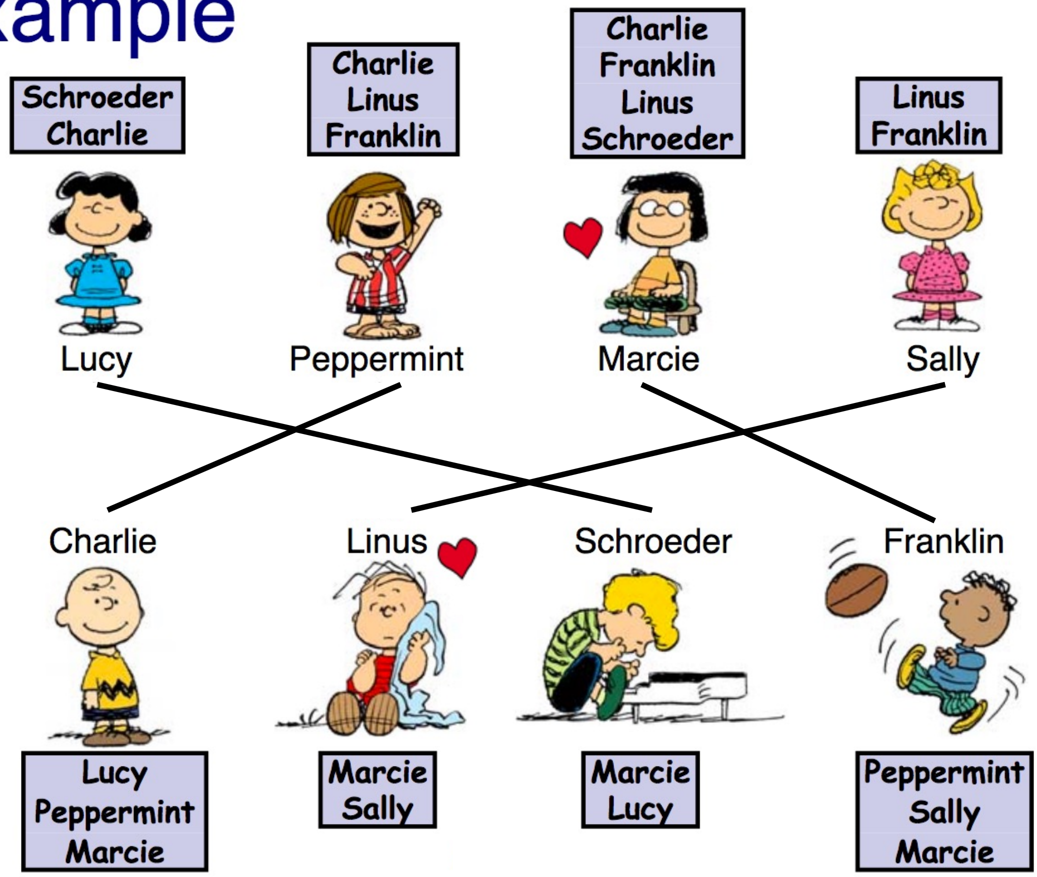
Example

- Alex: Betty Ann Cindy
- Bert: Ann Cindy Betty
- Carl: Ann Cindy Betty

- Ann: Bert Alex Carl
- Betty: Alex Carl Bert
- Cindy: Bert Alex Carl

- Matching (Alex, Ann), (Bert, Betty), (Carl, Cindy) is not stable because Alex and Betty prefer each other over their current partner
  - Alex and Betty are a blocking pair (i.e., a “rogue couple”)
- Stable matching: (Alex, Betty), (Bert, Ann), (Carl, Cindy)
  - Because a matching is stable if it has no blocking pair

Example



Stable marriage!

Original application

- Origin: assignment of med-school students to hospitals (for internships) for National Resident Matching Program in USA
  - Students list hospitals in order of preference
  - Hospitals list students in order of preference

- Gale-Shapley paper (1962): college admission and opposite-sex marriage
  - Students list colleges in order of preference
  - Colleges list students in order of preference

COLLEGE ADMISSIONS AND THE STABILITY OF MARRIAGE

D. GALE* AND L. S. SHAPLEY, Brown University and the RAND Corporation

1. Introduction. The problem with which we shall be concerned relates to the following typical situation: A college is considering a set of n applicants of which it can admit a quota of only q. Having evaluated their qualifications, the admissions office must decide which ones to admit. The procedure of offering admission only to the q best-qualified applicants will not generally be satisfactory, for it cannot be assumed that all who are offered admission will accept.

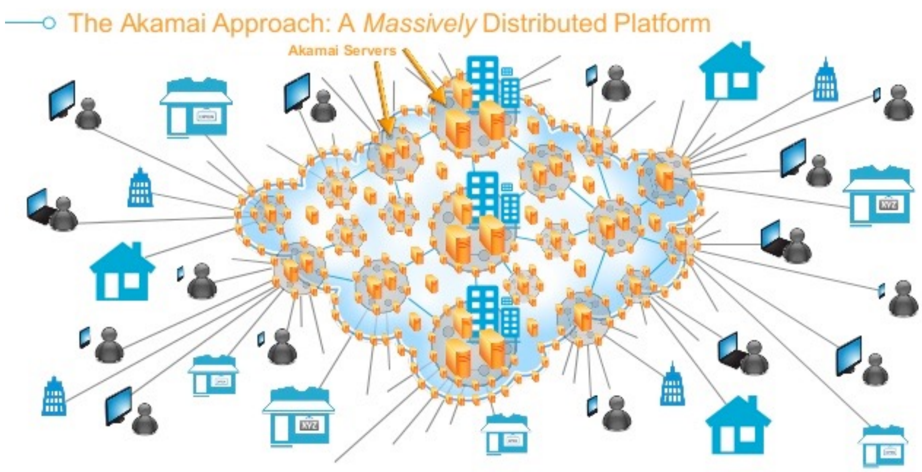
Practical applications

- A wide variety of practical applications, e.g.,:
  - Matching resident doctors to hospitals
  - Matching students to colleges
  - Matching sailors to ships
  - Matching kidneys and other organs with patients who require transplant
  - Matching users to servers in a large distributed Internet service (Akamai)

- More generally to any two-sided market
  - A market with 2 distinct groups of participants each with their own preferences

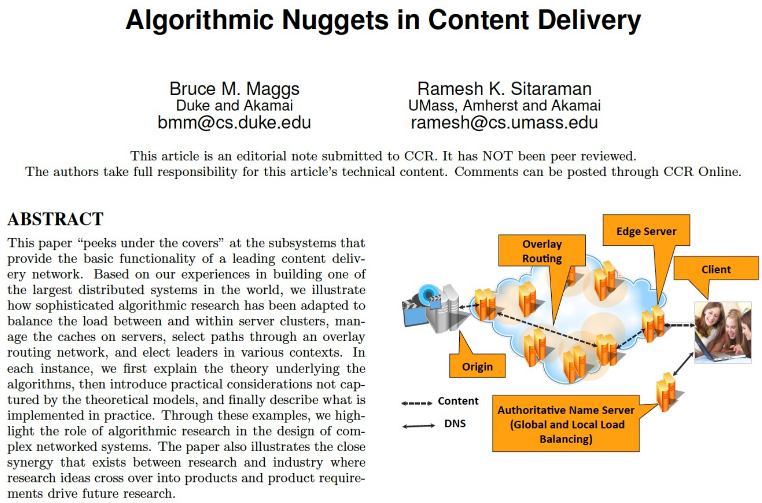
Break: what is Akamai?

- The largest content delivery network https://www.akamai.com/
- Distributes much of world's content on web: delivers 15-30% of Internet traffic every day
  - More than 350000 servers distributed in 134 countries



Akamai and stable marriage

- Users: preferences based on latency and packet loss
- Web servers: preferences based on costs of bandwidth and co-location
- Goal: assign billions of users to web servers, every 10 seconds



Algorithmic Nuggets in Content Delivery

Bruce M. Maggs
Duke and Akamai
bmm@cs.duke.edu

Ramesh K. Sitaraman
UMass, Amherst and Akamai
ramesh@cs.umass.edu

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

This paper "peeks under the covers" at the subsystems that provide the basic functionality of a leading content delivery network. Based on our experiences in building one of the largest distributed systems in the world, we illustrate how sophisticated algorithmic research has been adapted to balance the load between and within server clusters, manage the caches on servers, select paths through an overlay routing network, and elect leaders in various contexts. In each instance, we first explain the theory underlying the algorithms, then introduce practical considerations not captured by the theoretical models, and finally describe what is implemented in practice. Through these examples, we highlight the role of algorithmic research in the design of complex networked systems. The paper also illustrates the close synergy that exists between research and industry where research ideas cross over into products and product requirements drive future research.

Simple-minded approach 1

- "**Greedy**" approach:
  - Greedy for women: First woman chooses, then second woman chooses …
  - Greedy for men: First man chooses, then second man chooses …
- Claim: it does not work

Simple-minded approach 1: example

Preference list for men
- Alex: Ann Betty Cindy
- Bert: Ann Betty Cindy
- Carl: Betty Ann Cindy

Preference list for women
- Ann: Carl Bert Alex
- Betty: Carl Bert Alex
- Cindy: Bert Carl Alex

Alex: Ann Betty Cindy

Bert: ~~Ann~~ Betty Cindy

Carl: ~~Betty~~ ~~Ann~~ Cindy

Ann: Carl Bert Alex

Betty: Carl Bert Alex

Cindy: Bert Carl Alex

- Greedy matching for men (Alex, Ann), (Bert, Betty), (Carl, Cindy) is not stable
  - Ann and Carl prefer each other over assigned partner

Simple-minded approach 1: example

Preference list for men
- Alex: Ann Betty Cindy
- Bert: Ann Betty Cindy
- Carl: Betty Ann Cindy

Preference list for women
- Ann: Carl Bert Alex
- Betty: Carl Bert Alex
- Cindy: Bert Carl Alex

Alex: Ann Betty Cindy

Bert: Ann Betty Cindy

Carl: Betty Ann Cindy

Ann: Carl Bert Alex

Betty: ~~Carl~~ Bert Alex

Cindy: ~~Bert~~ ~~Carl~~ Alex

- Greedy matching for women (Ann, Carl), (Betty, Bert), (Cindy, Alex) is not stable
  - Betty and Carl prefer each other over assigned partner

What is a greedy algorithm?

- A greedy algorithm is a simple, intuitive algorithm used to solve optimization problems
- It makes the optimal choice at each step as it attempts to find the overall optimal solution to the entire problem

Simple-minded approach 2

- "**Local search**" approach
  - Call it Soap-Series-Algorithm

    While there is a blocking pair

    Do switch the blocking pair

- Claim: it can go on forever (so it gives no solution for some input)

Simple-minded approach 2: example

- Consider the following preference list

w_1	m_2	m_3	m_1	m_1	w_1	w_3	w_2
w_2	m_1	m_2	m_3	m_2	w_2	w_1	w_3
w_3	m_3	m_1	m_2	m_3	w_1	w_2	w_3

Simple-minded approach 2: example

- Consider the following preference list
- Start from this matching

w_1	m_2	m_3	m_1	m_1	w_1	w_3	w_2
w_2	m_1	m_2	m_3	m_2	w_2	w_1	w_3
w_3	m_3	m_1	m_2	m_3	w_1	w_2	w_3

- w_1 & m_3 run away together: switch

Simple-minded approach 2: example

- Consider the following preference list
- Match w_1 and m_3

w_1	m_2	m_3	m_1
w_2	m_1	m_2	m_3
w_3	m_3	m_1	m_2

m_1	w_1	w_3	w_2
m_2	w_2	w_1	w_3
m_3	w_1	w_2	w_3

- w_2 & m_1 run away together: switch

Simple-minded approach 2: example

- Consider the following preference list
- Match w_2 and m_1

w_1	m_2	m_3	m_1
w_2	m_1	m_2	m_3
w_3	m_3	m_1	m_2

m_1	w_1	w_3	w_2
m_2	w_2	w_1	w_3
m_3	w_1	w_2	w_3

- w_3 & m_1 run away together: switch

Simple-minded approach 2: example

- Consider the following preference list
- Match w_3 and m_1

w_1	m_2	m_3	m_1
w_2	m_1	m_2	m_3
w_3	m_3	m_1	m_2

m_1	w_1	w_3	w_2
m_2	w_2	w_1	w_3
m_3	w_1	w_2	w_3

- w_1 & m_2 run away together: switch

Simple-minded approach 2: example

- Consider the following preference list
- Match w_1 and m_2

w_1	m_2	m_3	m_1
w_2	m_1	m_2	m_3
w_3	m_3	m_1	m_2

m_1	w_1	w_3	w_2
m_2	w_2	w_1	w_3
m_3	w_1	w_2	w_3

- w_2 & m_2 run away together: switch

Simple-minded approach 2: example

- Consider the following preference list
- Match w_2 and m_2

w_1	m_2	m_3	m_1
w_2	m_1	m_2	m_3
w_3	m_3	m_1	m_2

m_1	w_1	w_3	w_2
m_2	w_2	w_1	w_3
m_3	w_1	w_2	w_3

- w_3 & m_3 run away together: switch

Simple-minded approach 2: example

- Consider the following preference list
- Match w_3 and m_3

w_1	m_2	m_3	m_1
w_2	m_1	m_2	m_3
w_3	m_3	m_1	m_2

m_1	w_1	w_3	w_2
m_2	w_2	w_1	w_3
m_3	w_1	w_2	w_3

- This was exactly the starting point!

Simple-minded approach 2

- "Local search" approach does not need to terminate

    While there is a blocking pair

    Do switch the blocking pair

- Can go on forever
- We need something else

Local search algorithms

- Local search algorithms move from solution to solution in the space of candidate solutions (the *search space*) by applying local changes, until a solution deemed optimal is found or a time bound is reached

Gale-Shapley algorithm

- The algorithm always finds a stable matching
  - Input: a list of men and women and their preference lists
  - Output: a stable matching
- Note: we will not go over the original description, which was not gender-neutral
  - Still not completely politically correct ☺
    - First, we will see the women-optimal solution (and then the men-optimal solution), where women propose
  - The algorithm is biased towards the proposing side

Gale-Shapley algorithm

- Fix an ordering of the women
- Repeat until everyone is matched:
  - Let A be the first unmatched woman in the ordering
  - Find a man B such that B is the most desirable man in A's list who is either unmatched, or currently matched to a woman C but B prefers A to C
  - Match A and B; if B was previously matched to C, then C becomes unmatched

Gale-Shapley algorithm: example

Input:

Preference list for men

- Alex: Ann Betty Cindy
- Bert: Ann Betty Cindy
- Carl: Betty Ann Cindy

Preference list for women

- Ann: Carl Bert Alex
- Betty: Carl Bert Alex
- Cindy: Bert Carl Alex

Let's apply Gale-Shapley algorithm

Ordering: Ann, Betty, Cindy

Pick Ann, match (Ann, Carl)

Ordering: ~~Ann~~, Betty, Cindy

Pick Betty, match (Betty, Carl) unmatching Ann

Ordering: Ann, ~~Betty~~, Cindy

Pick Ann, match (Ann, Bert)

Ordering: ~~Ann~~, ~~Betty~~, Cindy

Pick Cindy, match (Cindy, Alex)

Ordering: ~~Ann~~, ~~Betty~~, ~~Cindy~~

Stable matching: (Ann, Bert), (Betty, Carl), (Cindy, Alex)

Gale-Shapley algorithm

- Does this algorithm terminate?
- How fast?
- Does this algorithm give a stable matching?

Termination and number of steps

- Once a man is matched, he never becomes unmatched, although his partner may change
  – Women can switch between being free and engaged
- When a man's partner changes, it is always to a woman he prefers more: this can happen at most *n*-1 times
  – Recall that *n* is the number of women (and men)
- Since there are *n* men, the total number of such events is at most n^2
- Note that n^2 is also the total input size
  – *n* women and *n* men, each with a preference list of length n

Stability of final matching

- Proof *by contradiction*
- Suppose final matching is not stable and there exists a blocking pair (Alex, Betty)
- Assume:
  – Alex is matched with Ann
  – Bert is matched with Betty
  – Alex prefers Betty to Ann
  – Betty prefers Alex to Bert
- Since Alex prefers Betty to Ann, Betty appears before Ann in Alex's list. Therefore, Betty must have proposed to Alex before proposing to Bert

Two cases:

1. When Betty proposes to Alex, he has a partner (say Carola) preferable to Betty: Carola is also preferable to Ann, but men can get only more preferable partners → contradiction
2. When Betty proposes to Alex, he is free, but Betty is later replaced for a more preferred partner. Again contradiction, Alex can never end up with Ann

Women-optimal stable matching

- Theorem
  - All possible executions of Gale-Shapley algorithm give the same stable matching
  - In this matching, the women get the best partner they can have in any stable matching
  - In this matching, the men get the worst partner they can have in any stable matching

Men optimal stable matching

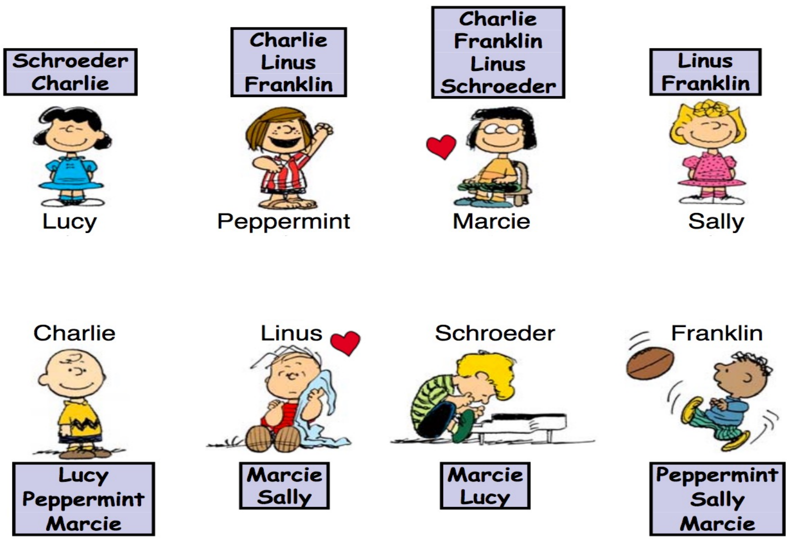
- For gender balance ☺
- In slide 29, let's interchange the roles of women and men so to achieve *men optimal* stable matching
  - In this matching, the men get the best partner they can have in any stable matching
  - In this matching, the women get the worst partner they can have in any stable matching

Take-away

- Stable matching always exists
- The simple-minded soap-opera algorithm does not necessarily terminate
  - "Start with any matching, and make switches when a pair prefers each other to their current partner"
- A stable matching can be found efficiently (*linear* in the input size) using the Gale-Shapley algorithm
- Controversy: women-optimal Gale-Shapley algorithm is better for women (hospitals in the original application)

Exercise 1

- Find a stable marriage for the Peanuts



Exercise 2

- Set of women W = $\{w_1, w_2, w_3\}$
- Set of men M= $\{m_1, m_2, m_3\}$
- Preference lists:
  - m_1: w_1 w_2 w_3 (i.e., m_1 prefers w_1 to w_2 to w_3)
  - m_2: w_2 w_1 w_3
  - m_3: w_3 w_2 w_1
  - w_1: m_1 m_2 m_3
  - w_2: m_3 m_1 m_2
  - w_3: m_2 m_1 m_3
- Find the women-optimal stable marriage and men-optimal stable marriage
  - Do these matchings differ?
  - Does the order of unmatched women/men affect the stable matching you have found?

References

- Shapley and Gale, College admissions and the stability of marriage, *The American Mathematical Monthly*, 69(1):9–15, 1962. https://sites.math.washington.edu/~billey/classes/562.winter.2018/articles/Gale.Shapley.pdf
- Wikipedia, Stable marriage problem https://en.wikipedia.org/wiki/Stable_marriage_problem
- The Royal Swedish Academy of Science, Stable matching: Theory, evidence, and practical design, 2012. https://www.nobelprize.org/uploads/2018/06/popular-economicsciences2012.pdf
- Meyer, Stable Matching Video, MIT 6.042J Mathematics for Computer Science, 2015. https://www.youtube.com/watch?v=RE5PmdGNgj0