

Lecture 6

CSE 331

Sep 12, 2022

2nd T/F poll up

poll #71   

stop following **15 views**

Actions *

2nd T/F poll

Is the following statement true or false:

In every Stable Matching problem instance where a man m and woman w have each other as their least preferred partner, the following is true. There is no stable matching for the instance where (m, w) are matched.

(Note by a stable matching problem instance, we mean both the set of men and women as well as all the $2n$ preference lists.)

True
 False

Submit

You have not yet voted.

Re-voting is not allowed. Select your vote and click submit to register your vote.

Your name will not be visible to anyone.

[t/f_poll](#) [polls](#)

We're not mind readers



If you need it, ask for help



Make sure you can run HW0 code

note #72   

[stop following](#) **1 view** [Actions](#)

Make sure you can run HW 0 template

If you did not submit [HW 0 Q3](#), please make sure you have the setup on your computer so that you can run the HW 0 Q3 template code (in whichever language you prefer). If you need it, please go to an OH on Monday or Tuesday in case you cannot get things to setup and need help.

Note that the [autolab page](#) has instructions on how to setup the template code in IntelliJ (so only for java). By Wed, we hope to post a video on how to run the template code from command line on a VM.

Please note that after Wednesdays, the office hours will give preference to questions specifically about HW 1 and not questions on setup (like making sure you have your IDE/compiler ready to go).

[homework0](#) [autolab](#) [office_hours](#)

[Edit](#) good note | 0 Updated 11 seconds ago by AM Rude

Register your project groups

Deadline: Friday, Sep 30, 11:59pm

CSE 331

Syllabus

Piazza

Schedule

Homeworks -

Autolab

Project -

Support Pages -

channel

Sample Exams -

Forming groups

You form groups of size **exactly three (3)** for the project. Below are the various logs

Project Overview

Group signup form

• You have two choices in forming your group:

1. You can form your group on your own: i.e. you can submit the list of **EXACTLY** three (3) groups members in your group.

</> Note

Note that if you pick this option, your group needs to have **exactly THREE (3)** members. In particular, if your group has only two members you cannot submit as a group of size two. If you do not know many people in class, feel free to use piazza to look for the third group member.

Also, if you form a group of size three, please make **only one submission per group**.

2. You can submit *just your name*, and you will be assigned a random group among *all* students who take this second option. However, **note that if you pick this option you could end up in a group of size 2**. There will be at most two groups of size 2.

</> Potential risk

Note that if you pick the option of being assigned a random group, you take on the risk that a assigned group might not "pull their weight." We unfortunately cannot help with such aspects of group dynamics. (Of course if a group member is being abusive, please do let Atri know.) Please note that a group member who does not do much work will get penalized on the [individual component](#) of the project grade.

Submitting your group composition

Use [this Google form](#) to submit your group composition (the form will allow you to pick one of the two options above).

• You need to fill in the form for group composition by **11:59pm on Friday, September 30**.

</> Deadline is strict!

If you do not fill in the form for group composition by the deadline, then you get a zero for the entire project.

HW 1 gets released this Tue

Mon, Sep 12	Gale Shapley algorithm     x^4	[KT, Sec 1.1] Reading Assignment: Pigeonhole principle Reading Assignment: Asymptotic notation care package
Tue, Sep 13		(HW 1 out)
Wed, Sep 14	Gale Shapley algorithm outputs a stable matching     x^4	[KT, Sec 1.1] Reading Assignment: Proof details of GS termination
Fri, Sep 16	Efficient algorithms and asymptotic analysis     x^4	[KT, Sec 1.1] Reading Assignment: Worst-case runtime analysis notes Reading Assignment: [KT, Sec 1.1, 2.1, 2.2, 2.4]
Mon, Sep 19	Runtime Analysis of Gale-Shapley algorithm     x^4	[KT, Sec 2.3]
Tue, Sep 20		(HW 2 out, HW 1 in)
Wed, Sep 21	Graph Basics     x^4	[KT, Sec 2.3, 3.1]
Fri, Sep 23	Computing Connected Component     x^4	[KT, Sec 3.2] Reading Assignment: Care package on trees Reading Assignment: BFS by examples
Mon, Sep 26	Explore Algorithm     x^4	[KT, Sec 3.2]
Tue, Sep 27		(HW 3 out, HW 2 in)
Wed, Sep 28	Runtime Analysis of BFS algorithm     x^4	[KT, Sec 3.3]
Fri, Sep 30	More graph stuff     x^4	[KT, Sec 3.3, 3.6] (Project Team Composition Due) Reading Assignment: [KT, Sec 3.3, 3.4, 3.5, 3.6] Reading Assignment: Care package on topological ordering
Mon, Oct 3	Interval Scheduling Problem     x^4	[KT, Sec 4.1]

Reading Assignment - I

note 073 stop following 3 views Actions

Reading Assignment: Asymptotic Analysis

As one of the changes made in F19, we will assume that y'all are familiar with asymptotic analysis and not spend reviewing it in any detail during the lectures. In case you are not that comfortable with asymptotic analysis and/or want to review the material, please read through the asymptotic analysis case package:

<http://www.student.cse.buffalo.edu/~at1/cse321/support/case-package/asymptotics/index.html>

We will need this either the middle of lecture on Wednesday or in the Friday lecture.

lectures

good note

Updated 1 minute ago by Ash Rubin

Reading Assignment - II

note #74

stop following 1 view

Actions

Reading Assignment: Pigeonhole principle

Another reading assignment for this week (here is the other one: #73). Please go through this support page on pigeonhole principle--

<http://www-student.cse.buffalo.edu/~atri/cse331/support/pigeon/index.html>

It's actually a very simple result that turns out to be surprisingly powerful. We'll use this in the Mon/Wed lecture.

lectures

Edit good note 0

Updated 18 seconds ago by Atri Rudra

Questions/Comments?



Stable Marriage problem

Set of men M and women W

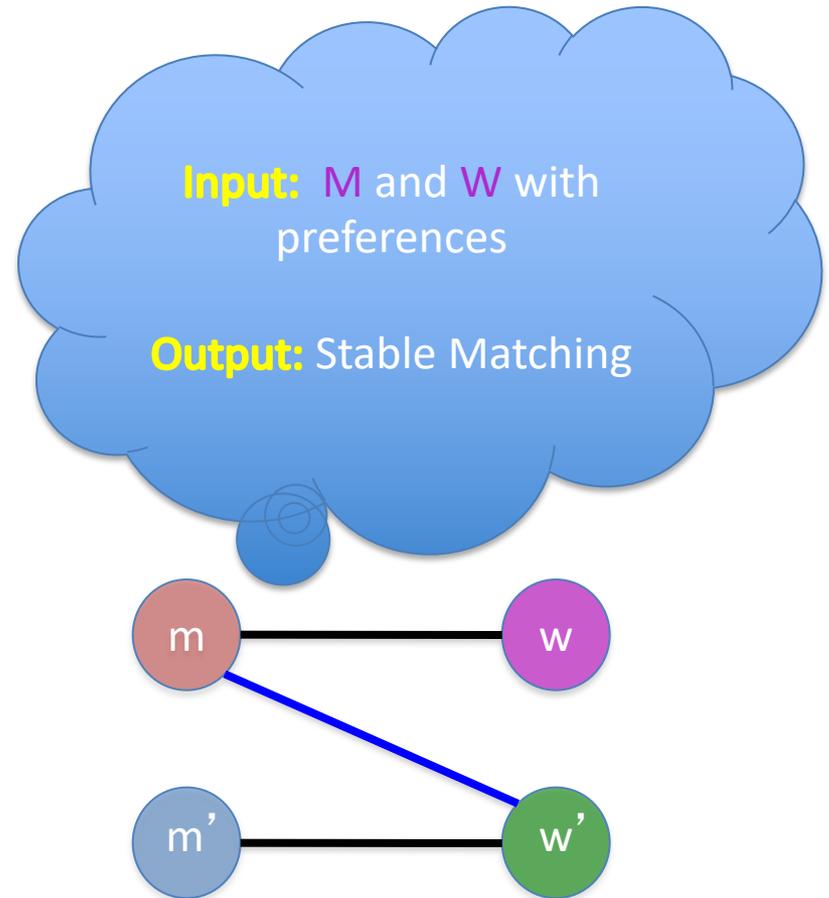
Preferences (ranking of potential spouses)

Matching (no polyandry/gamy in $M \times W$)

Perfect Matching (everyone gets married)

Instability

Stable matching = perfect matching + no instability



Two Questions

Does a stable marriage always exist?

If one exists, how quickly can we compute one?

The naïve algorithm

Incremental algorithm to produce all $n!$ perfect matchings?

Go through all possible perfect matchings S

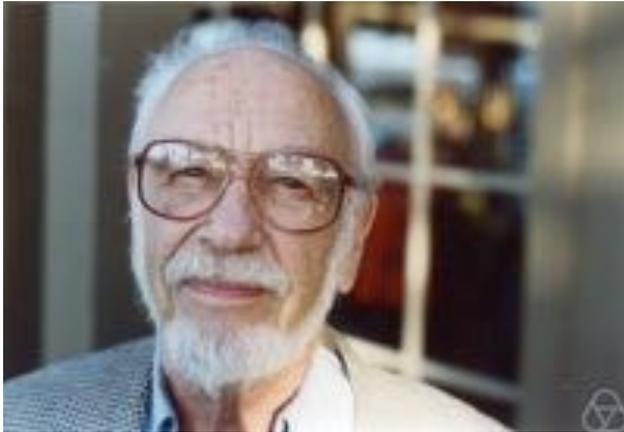
If S is a stable matching

then Stop



Else move to the next perfect matching

Gale-Shapley Algorithm



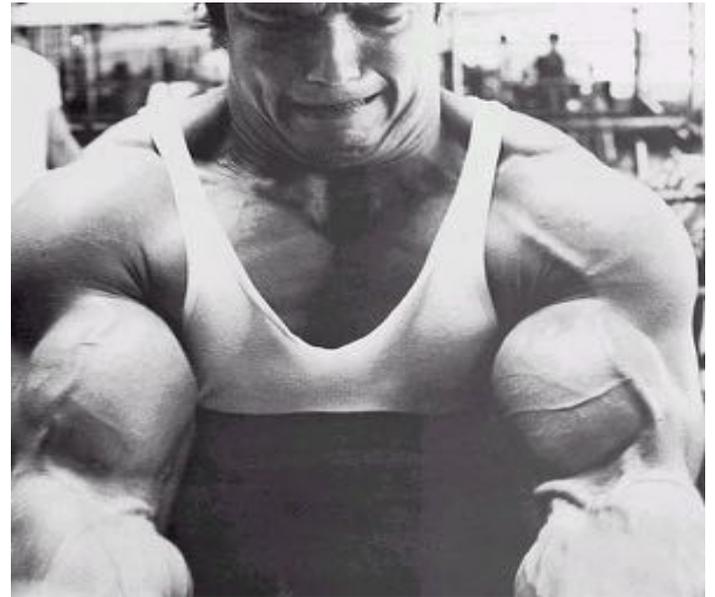
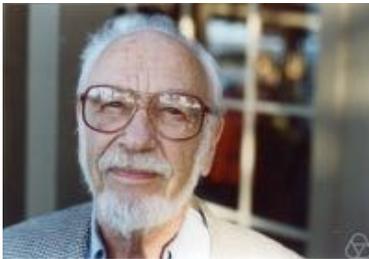
David Gale



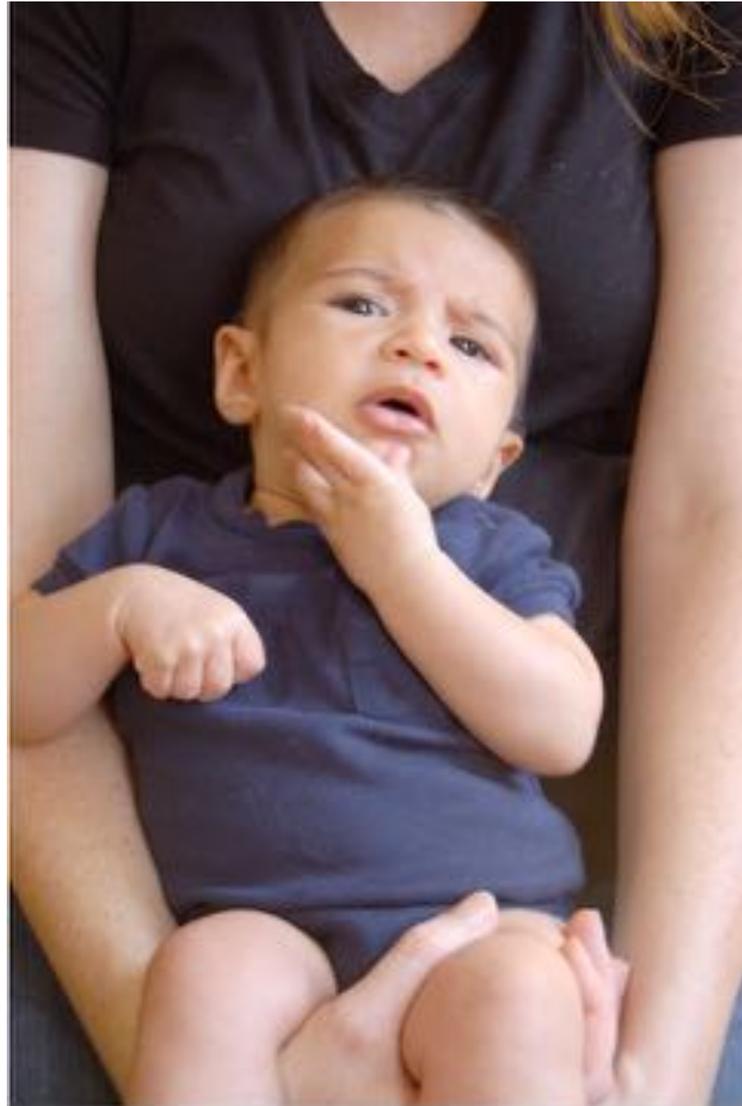
Lloyd Shapley

$O(n^2)$ algorithm

Moral of the story...



Questions/Comments?



Rest of today's agenda

Finish off GS algorithm

Run of GS algorithm on an instance

Prove correctness of the GS algorithm

Back to the board...



Gale-Shapley Algorithm

Initially all men and women are **free**

While there exists a free woman who can propose

Let w be such a woman and m be the best man she has not proposed to

w proposes to m

If m is free

(m,w) get **engaged**

Else (m,w') are engaged

If m prefers w' to w

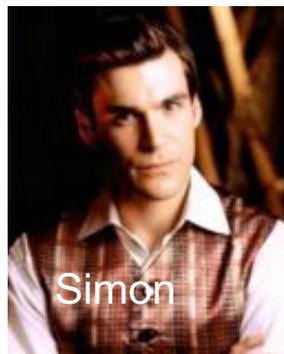
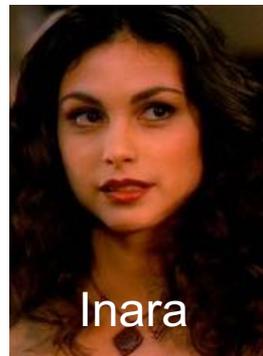
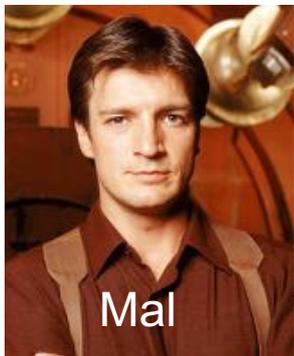
w remains **free**

Else

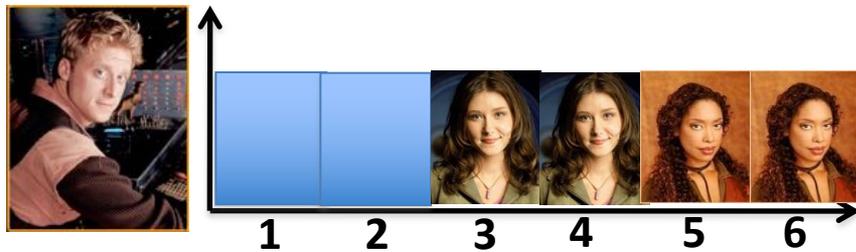
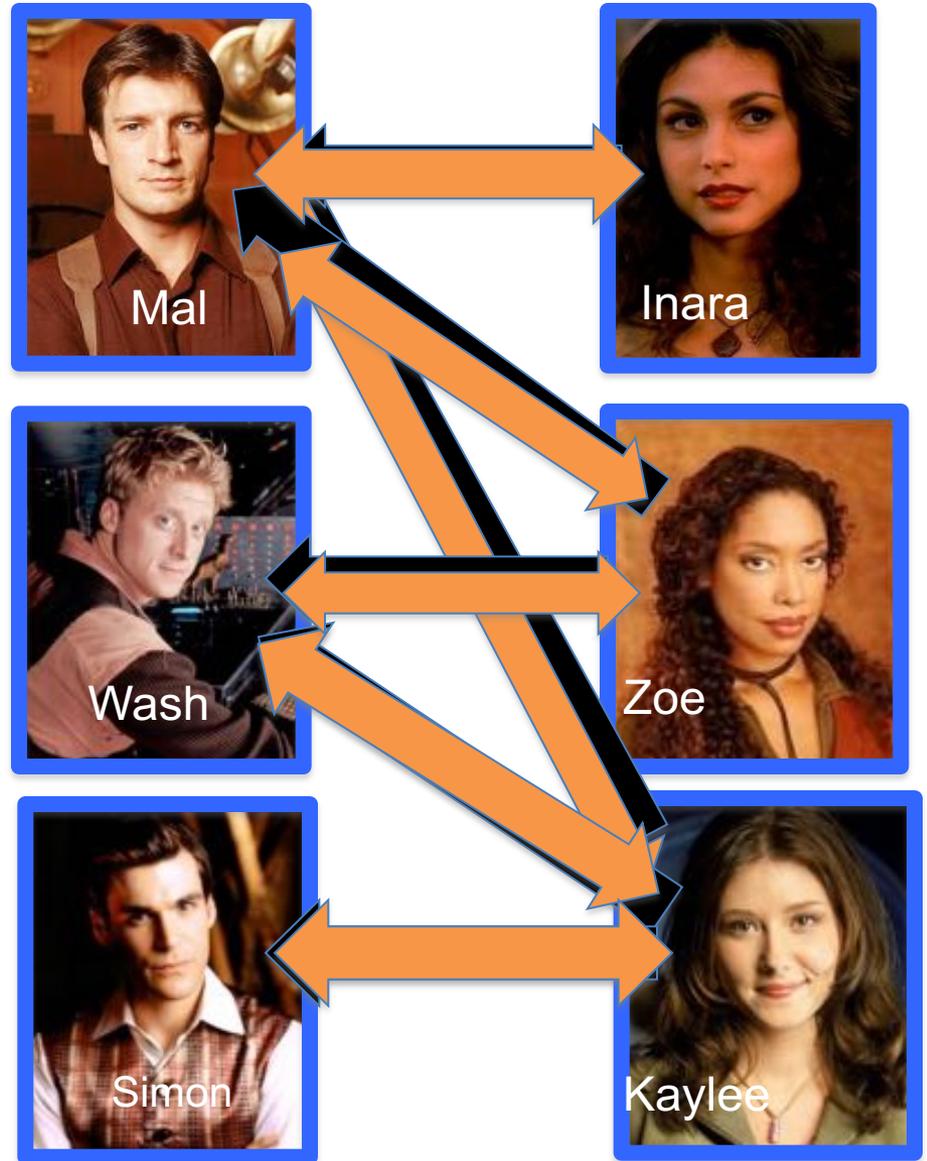
(m,w) get **engaged** and w' is **free**

Output the engaged pairs as the final output

Preferences



GS algorithm: Firefly Edition



Observation 1

Initially all men and women are **free**

While there exists a free woman who can propose

Let w be such a woman and m be the best man she has not proposed to

w proposes to m

If m is free

(m,w) get **engaged**

Else (m,w') are engaged

If m prefers w' to w

w remains **free**

Else

(m,w) get **engaged** and w' is **free**

Once a man gets engaged, he remains engaged (to “better” women)

Output the engaged pairs as the final output

Observation 2

Initially all men and women are **free**

While there exists a free woman who can propose

Let w be such a woman and m be the best man she has not proposed to

w proposes to m

If m is free

(m,w) get **engaged**

Else (m,w') are engaged

If m prefers w' to w

w remains **free**

Else

(m,w) get **engaged** and w' is **free**

If w proposes to m after m' , then she prefers m' to m

Output the set S of engaged pairs as the final output

Questions/Comments?



Why bother proving correctness?

Consider a variant where any free man **or** free woman can propose

Is this variant any different? Can you prove it?

GS' does not output a stable marriage

