Lecture 4

CSE 331

Sep 5, 2018

Please do keep on asking Qs!

note

stop following 119 views

Great job asking Questions

Today's lecture was probably the most interactive lecture 1 of CSE 331 ever, which was awesome! I hope the trend will continue throughout the semester :-)

Also a fair number of you stopped by my office hour today, which is also great! Please make full use of them for the rest of the semester too.

lectures

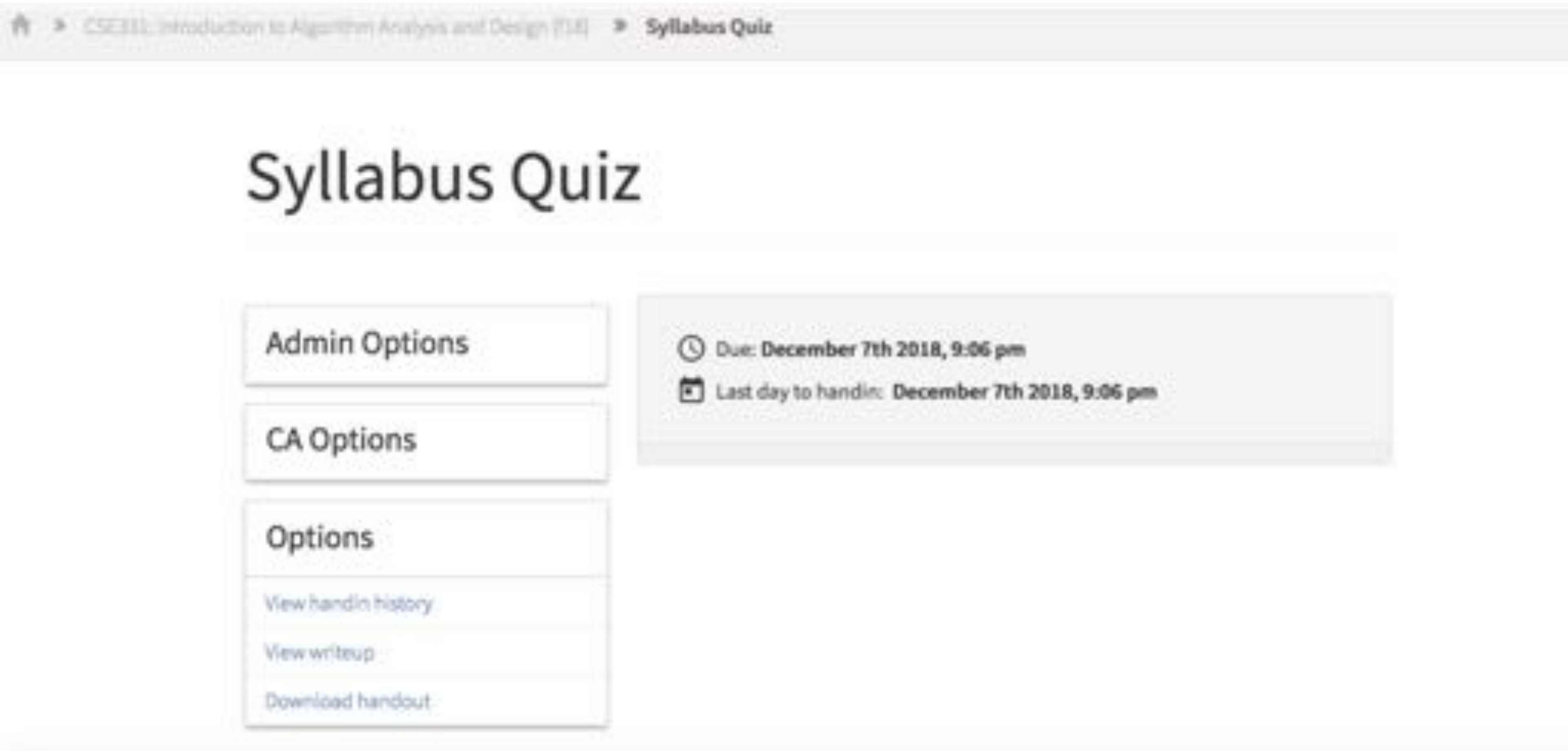
~ An instructor (Mehmet Ozdemir) thinks this is a good note ~

edit · good note | 1

Updated 1 day ago by Atri Rudra

Read the syllabus CAREFULLY!

No graded material will be handed back till you pass the syllabus quiz!



Separate Proof idea/proof details

</> Note

Notice how the solution below is divided into proof idea and proof details part. **THIS IS IMPORTANT: IF YOU DO NOT PRESENT A PROOF IDEA, YOU WILL NOT GET ANY CREDIT EVEN IF YOUR PROOF DETAILS ARE CORRECT.**

Proof Idea

As the hint suggests there are two ways of solving this problem. (I'm presenting both the solutions but of course you only need to present one.)

We begin with the approach of reducing the given problem to a problem you have seen earlier. Build the following complete binary tree: every internal node in the tree represents a "parent" RapidGrower while its two children are the two RapidGrowers it divides itself into. After s seconds this tree will have height s and the number of RapidGrowers in the container after s seconds is the number of leaf nodes these complete binary tree has, which we know is 2^s. Hence, the claim is correct.

The proof by induction might be somewhat simpler for this problem if you are not comfortable with reduction. In this case let $R(s)$ be the number of RapidGrowers after s seconds. Then we use induction to prove that $R(s) = 2^s$ while using the fact that $2 \cdot 2^s = 2^{s+1}$.

Proof Details

We first present the reduction based proof. Consider the complete binary tree with height s and call it $T(s)$. Further, note that one can construct $T(s+1)$ from $T(s)$ by attaching two children nodes to all the leaves in $T(s)$. Notice that the newly added children are the leaves of $T(s+1)$. Now assign the root of $T(0)$ as the original RapidGrower in the container. Further, for any internal node in $T(s)$ ($s \geq 0$), assign its two children to the two RapidGrowers it divides itself into. Then note that there is a one to one correspondence between the RapidGrowers after s seconds and the leaves of $T(s)$. Then we use the well-known fact (cite your 191/250 book here with the exact place where one can find this fact): $T(s)$ has 2^s leaves, which means that the number of RapidGrowers in the container after s seconds is 2^s, which means that the claim is correct.

TA office hours finalized

note

stop following 101 views

Actions

TA office hours finalized

You can find the TA office hours (starting from Tue, Sep 4) in the syllabus:

http://www-student.cse.buffalo.edu/~atri/cse331/fall18/policies/syllabus.html

(You can also see them in the 331 calendar, which is at the 331 webpage.)

I would like to draw your attention to two things:

- Some of the office hours are marked as "1-on-1"-- these are OH where you can sign up for 10 mins one-on-one dedicated slots.
    - See @71 for details.
- If you have a question on a specific language, please go to a TA who has that language listed for them.

#pin

office_hours

edit · good note | 0

Updated 2 days ago by Atri Rudra

1-on-1 appointments

CSE 331 | Syllabus | 1-on-1 meetings | Piazza | Schedule | Homeworks ▾ | Autolab | Mini Project ▾ | Support Pages ▾ | Youtube channel

Appointments

Instructions and important information for booking and canceling one-on-one meetings for CSE 331 Fall 2018.

This is a beta feature

We are rolling out one-on-one meetings for the first time in CSE 331 this fall so apologies in advance for all the bugs that we would need to iron out as the semester proceeds. If you spot a bug, please either post on piazza or email `cse-331-staff@buffalo.edu`. Thanks in advance for your patience and help!

Instructions for booking appointments

Follow these instructions to book one-on-one appointments with a TA (for a slot of 10 minutes).

1. Go to the course calendar and search for a desired meeting time slot. You can only pick the office hours that are marked as `1-on-1`.

We are starting off small

To start off with, we will have 30 slots per week. If this turns out to be popular, we will increase the number of one-on-one meeting slots later in the semester.

New TA: Dhruv

note

stop following 128 views

Actions

New TA: Dhruv Kumar

I'm happy to let y'all know that we will have another TA joining us this semester: Dhruv Kumar.

Dhruv's OH have not been finalized yet: will post when that happens (sometime this week).
#pin

office_hours logistics

edit good note | 1

Updated 2 days ago by Atri Rudra

Peer Notetaker Request

note

stop following | 65 views

Actions

Peer Notes Request

A student in your class is eligible for the services of a Peer Notetaker. Notetakers provide an essential service that helps ensure equal access to education for students who receive accommodations. Notetakers who qualify may also be paid a stipend by Accessibility Resources at the end of the semester. If you are interested in becoming a Peer Notetaker for this course, please contact 716-645-2608 or stu-notes@buffalo.edu as soon as possible. Notetakers are accepted on a first come, first serve basis.

(If you do end up volunteering for being a peer notetaker, please also let me know so that I know I do not have to send more reminders. --Atri)

#pin

logistics | lectures

edit | good note | 0

Updated 7 hours ago by Atri Rudra

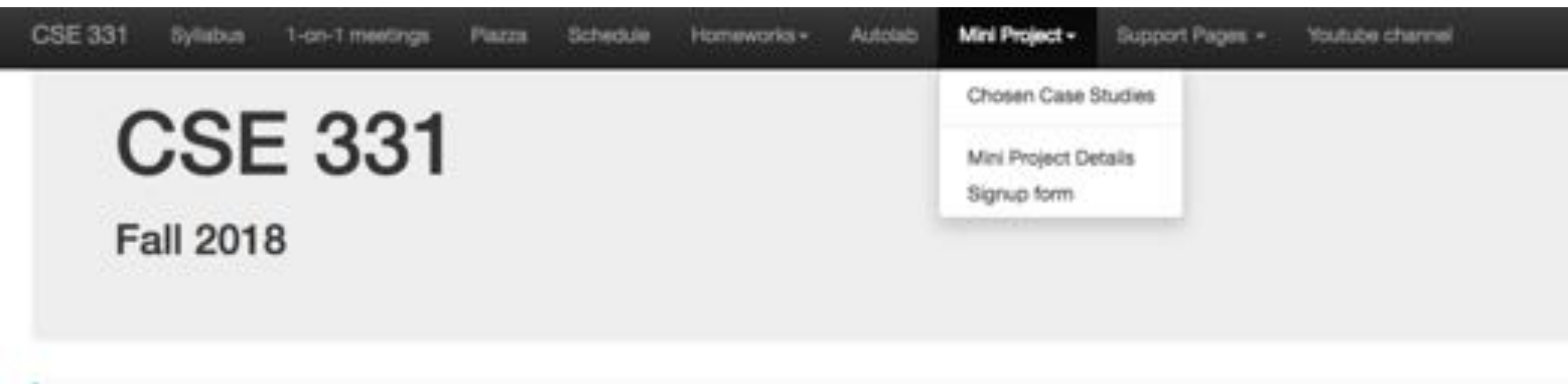
Makeup recitations

TODAY, 9-9:50am in Davis 338A

TODAY, 11-11:50am in Davis 338A

Sign-up for mini projects

Deadline: Monday, Sep 24, 11:59pm



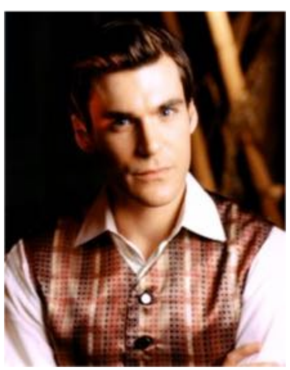
Questions/Comments?

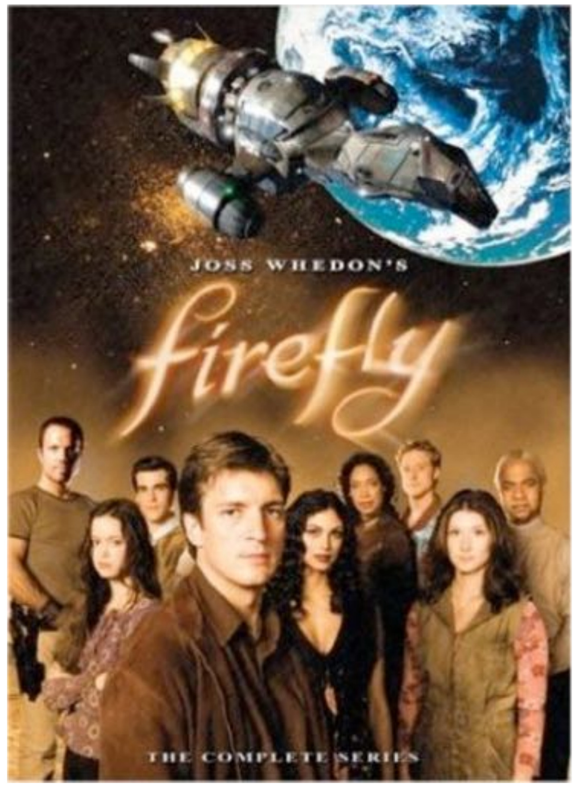
On matchings

Mal

Wash

Simon





Inara

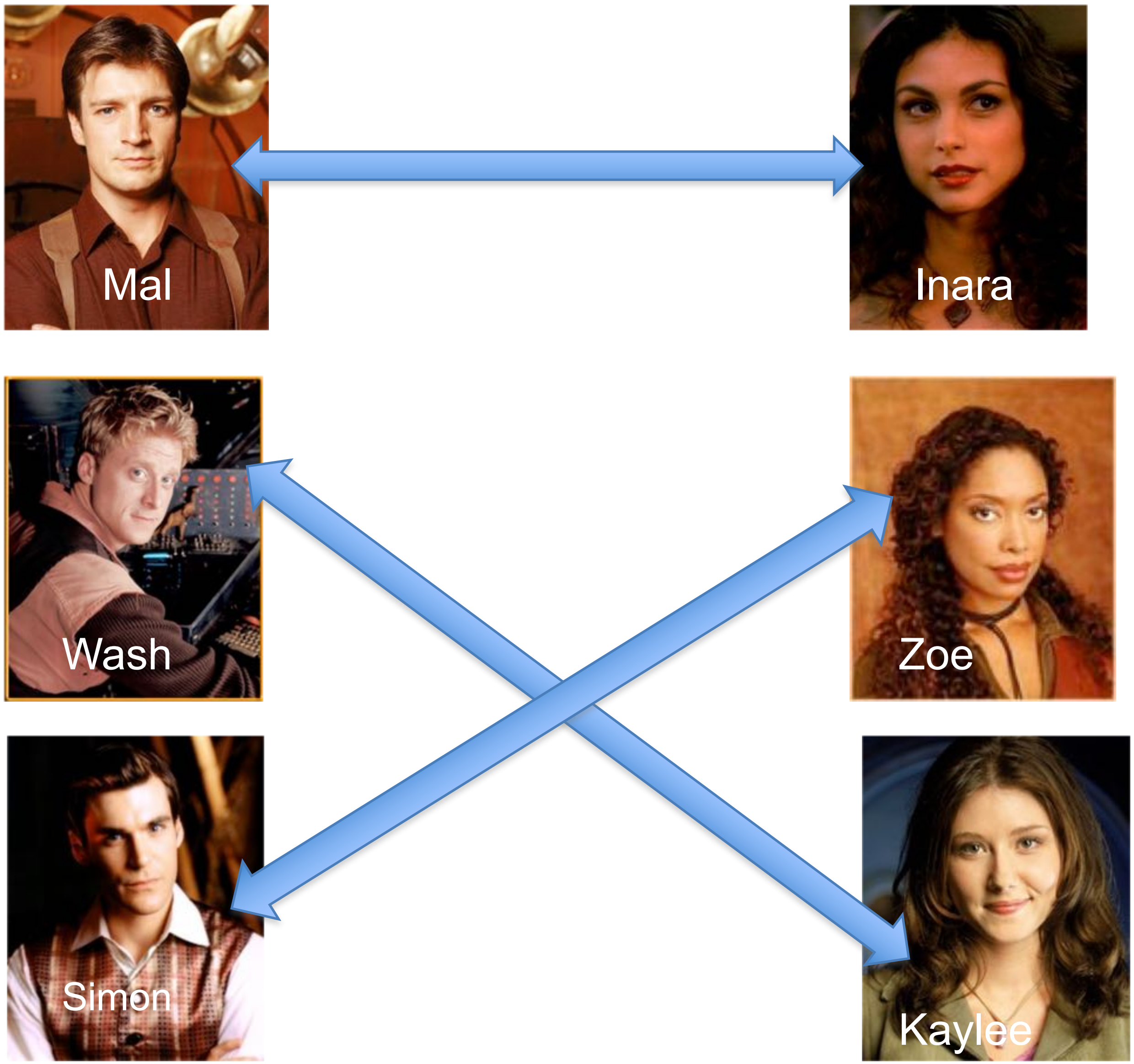
Zoe

Kaylee

A valid matching

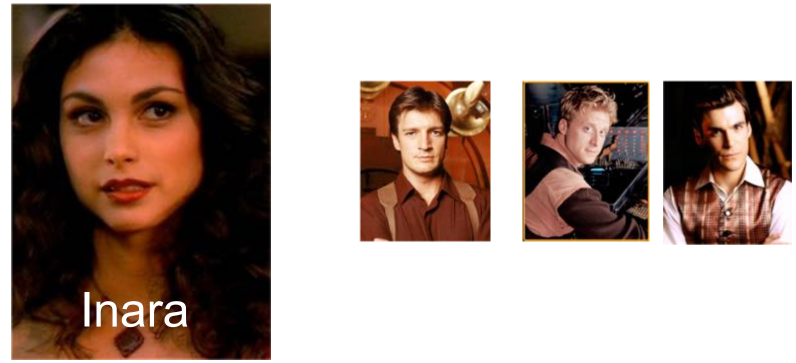
Not a matching

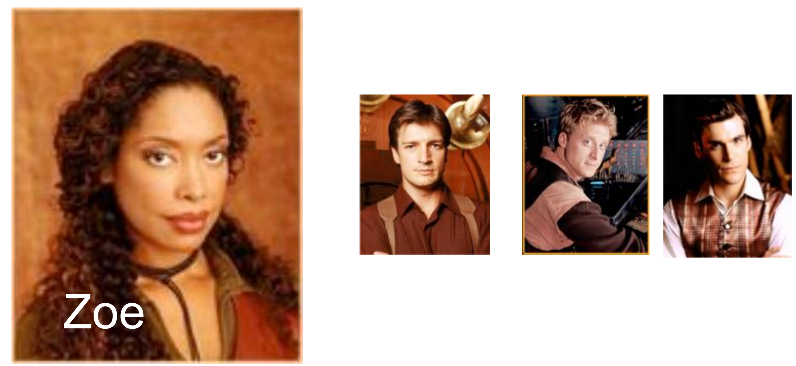
Perfect Matching

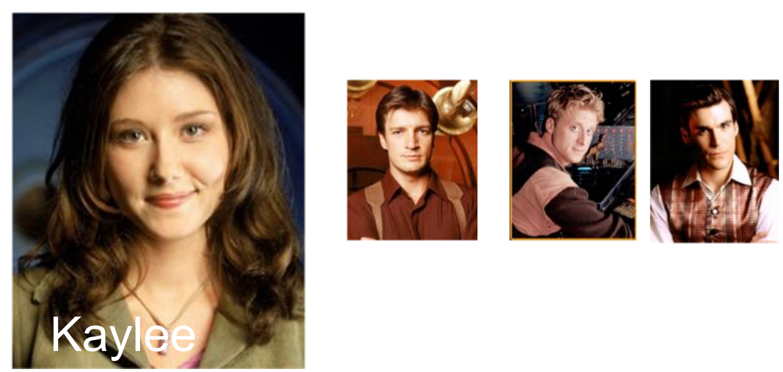


Back to couple more definitions

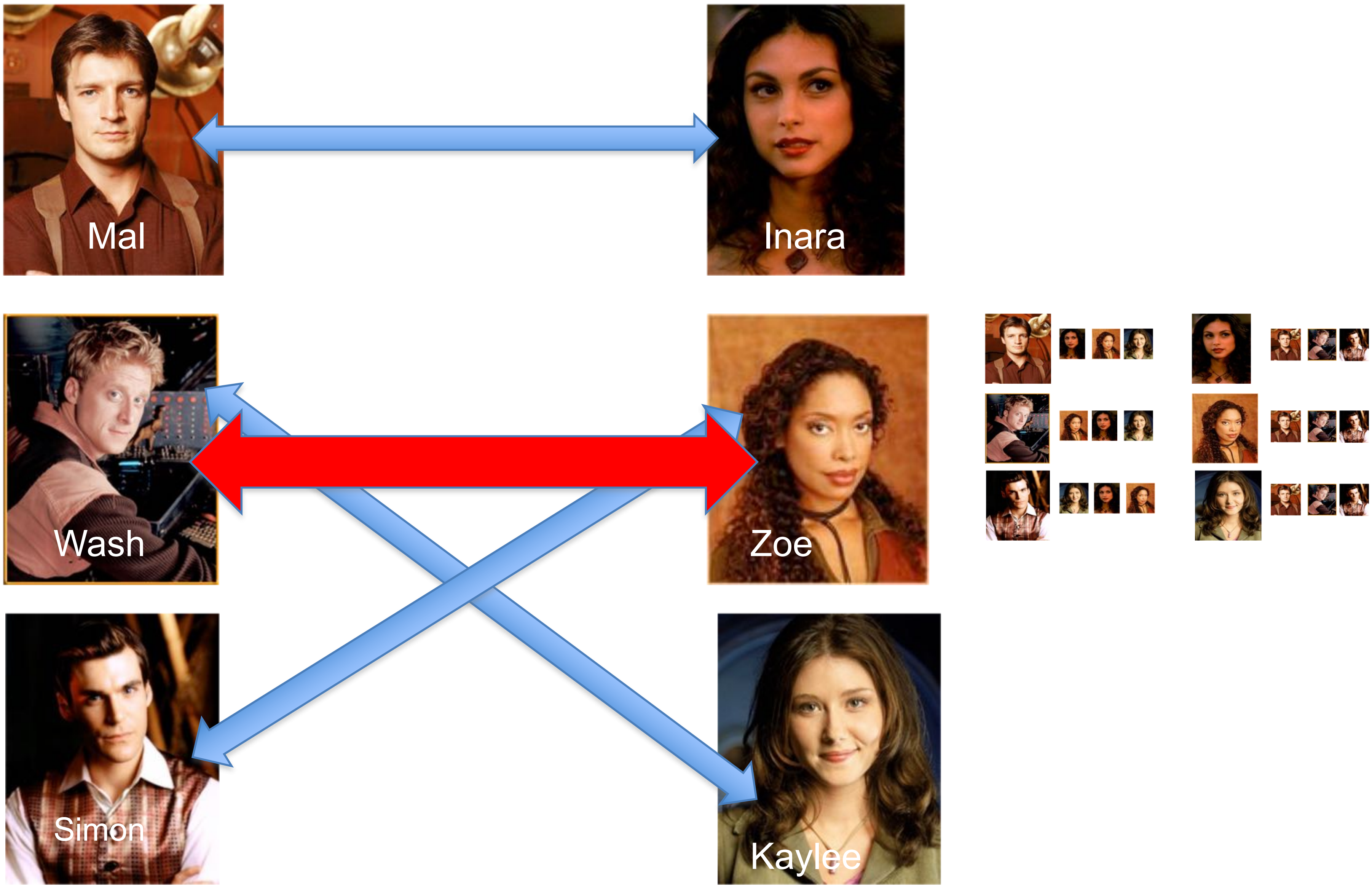
Preferences





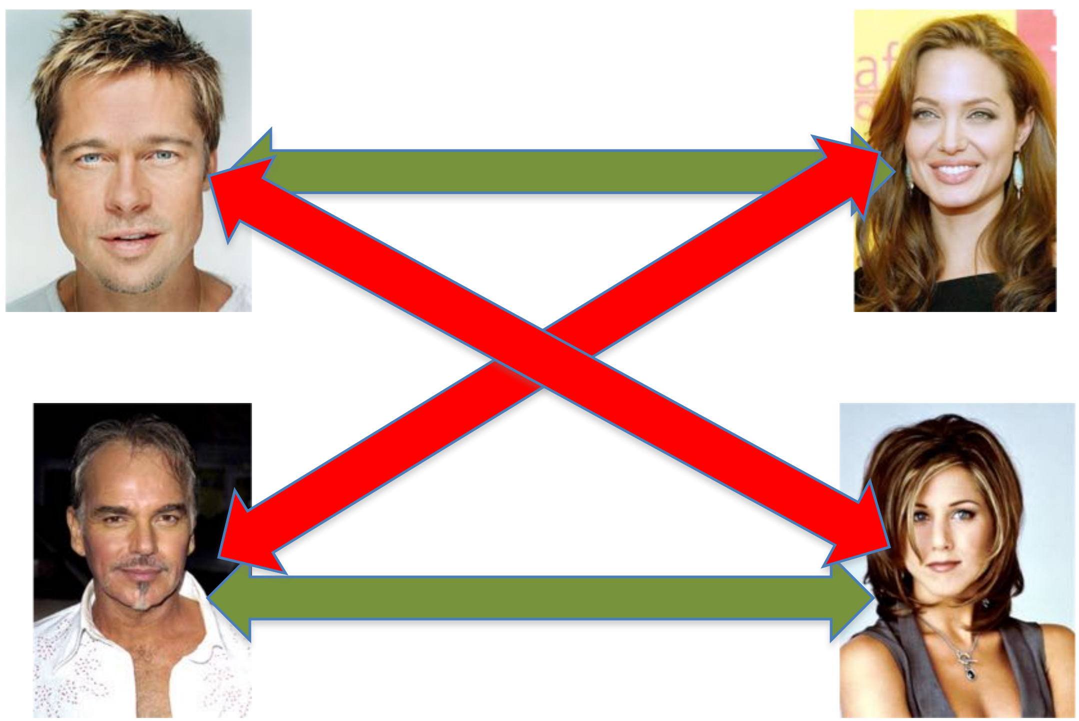


Instability

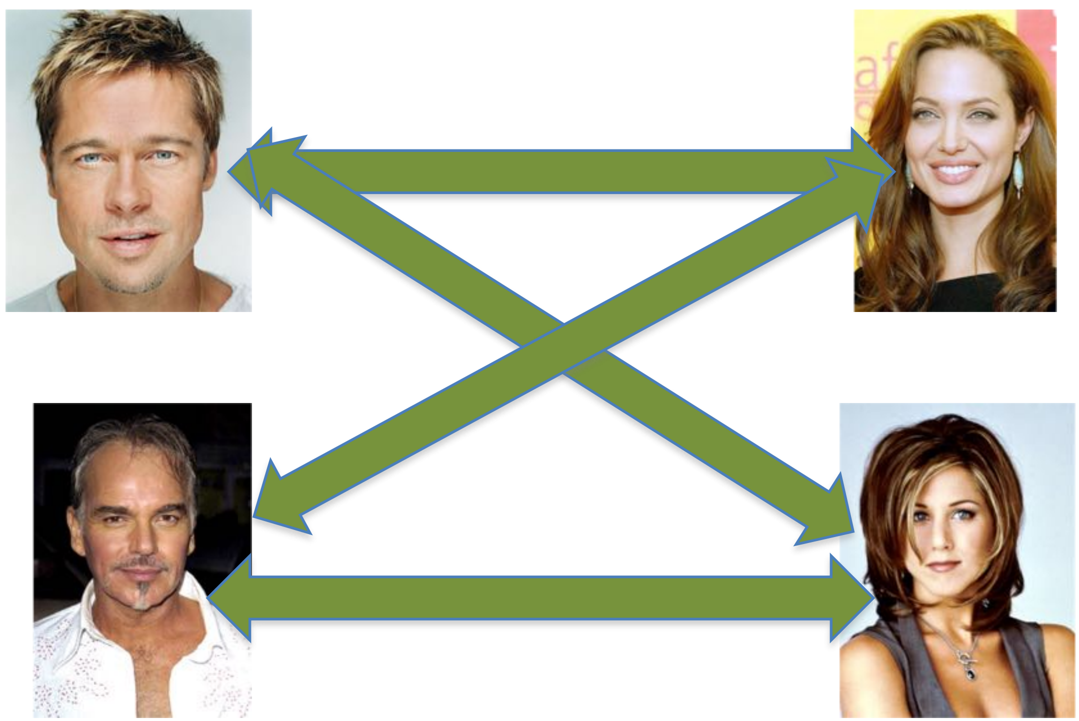


A stable marriage

Even though BBT and JA are not very happy



Two stable marriages



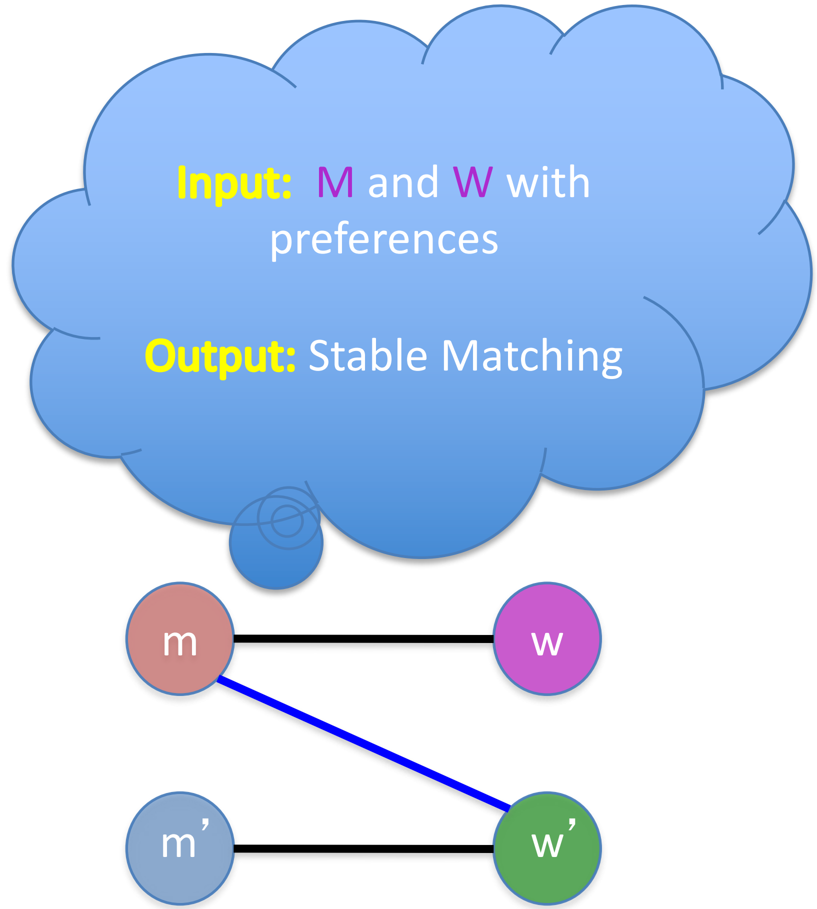
Stable Marriage problem

Set of men M and women W

Preferences (ranking of potential spouses)

Matching (no polyandry/gamy in M X W)

Perfect Matching (everyone gets married)



Instablity

Stable matching = perfect matching+ no instablity

Questions/Comments?

Two Questions

Does a stable marriage always exist?

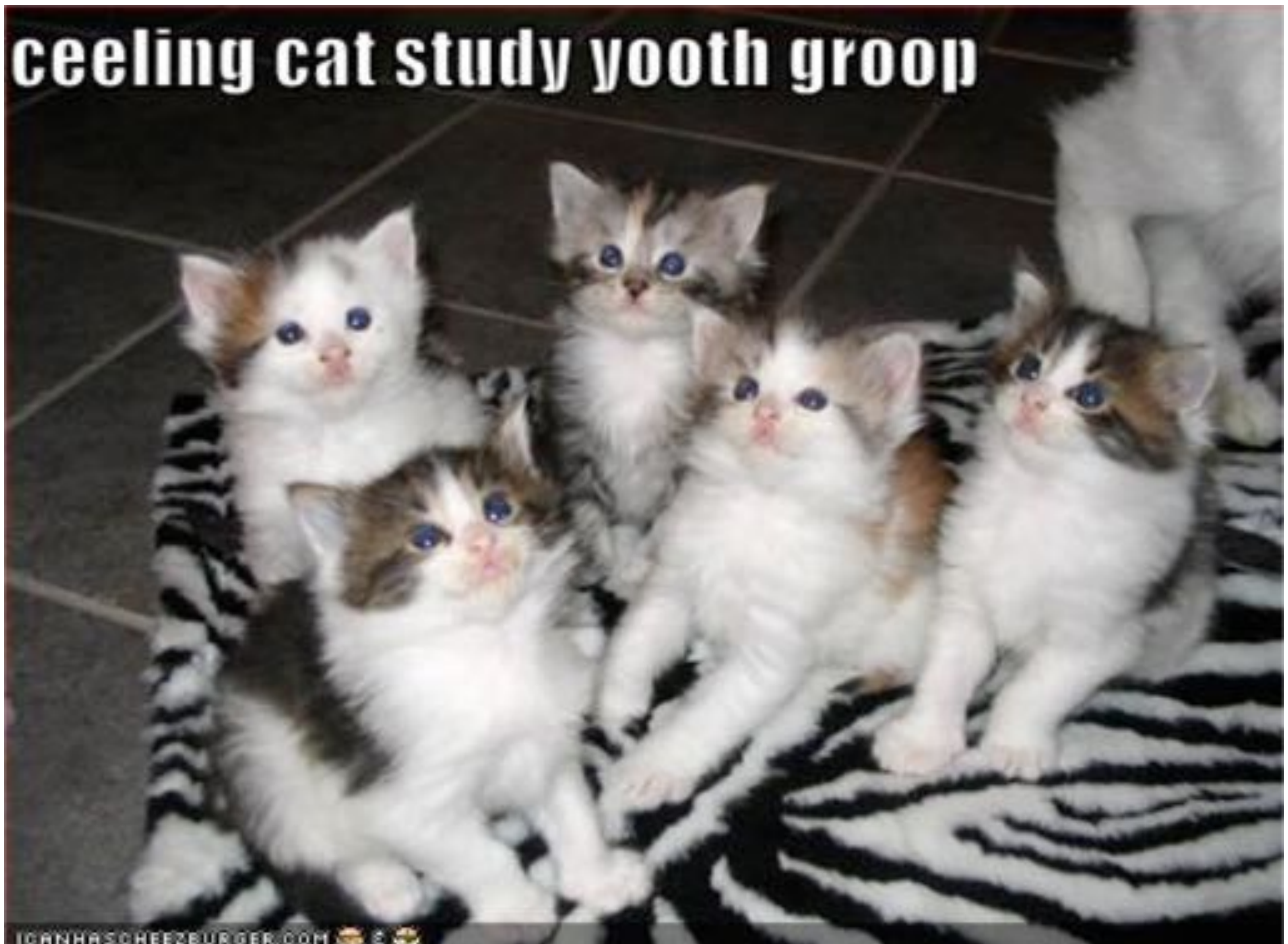
If one exists, how quickly can we compute one?

Today’s lecture

Naïve algorithm

Gale-Shapley algorithm for Stable Marriage problem

Discuss: Naïve algorithm!



The naïve algorithm

Incremental algorithm to produce all n! prefect matchings?

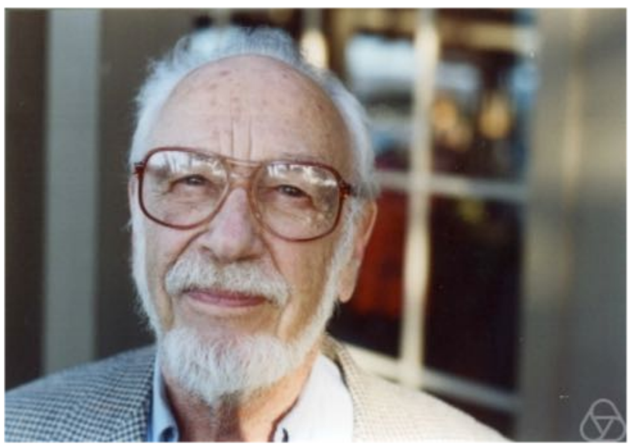
Go through all possible perfect matchings S

If S is a stable matching

then Stop

Else move to the next perfect matching

Gale-Shapley Algorithm

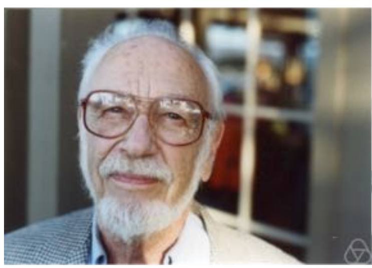


David Gale

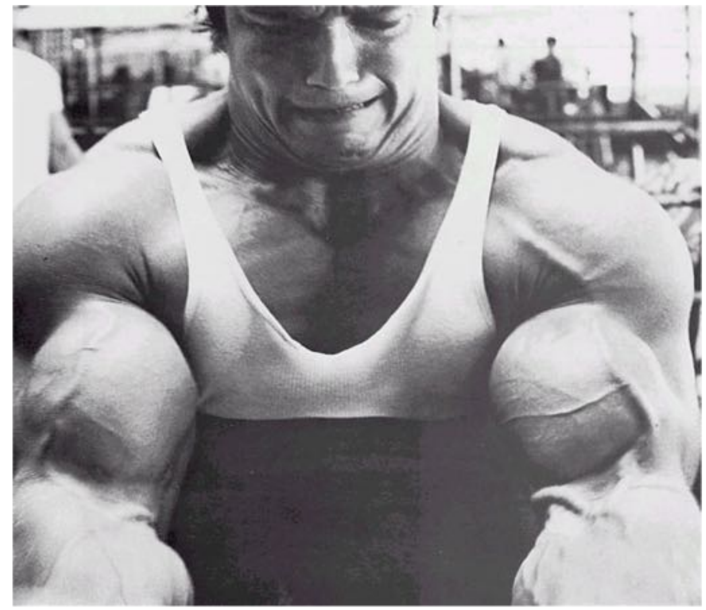
Lloyd Shapley

$O(n^3)$ algorithm

Moral of the story…



>



Questions/Comments?

Gale-Shapley Algorithm

Intially 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