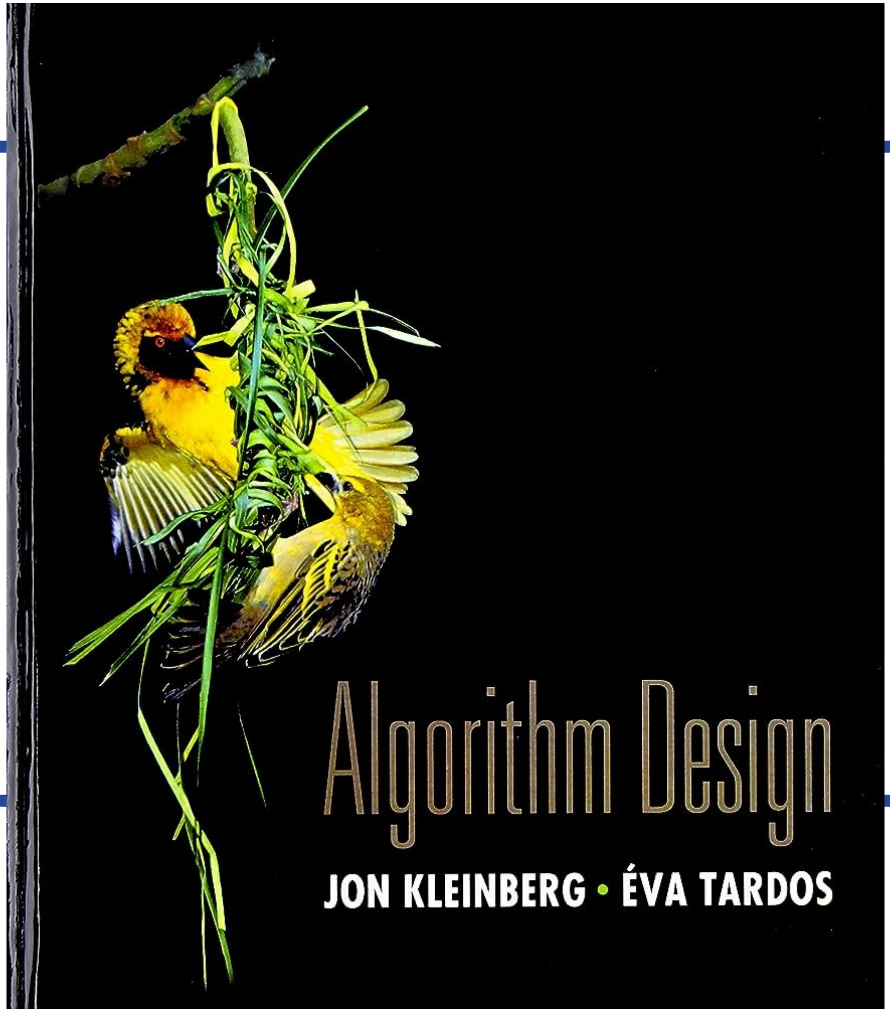
CS 310: Algorithms

Lecture 3

Instructor: Naveed Anwar Bhatti



Chapter 1:
Introduction and Some Representative Problems

Teaching Assistants – FINALIZED (S2)

Mohammad Jaffer Iqbal
24100064@lums.edu.pk
Thursday (12-1:30pm);
Friday (10am-11:30am)
Venue: Will mention venue in office hours email

Shehryar Khan
24100181@lums.edu.pk
Tuesday/Thursday 1:00 PM – 3:00 PM
Venue: Will mention venue in office hours email

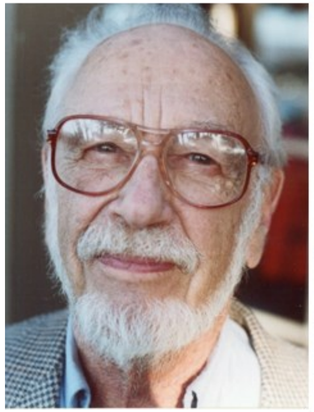
Aashish Jai
24100036@lums.edu.pk
M/W (2:30 - 3:30). Friday (By request)
Venue: Varying (Will Email)

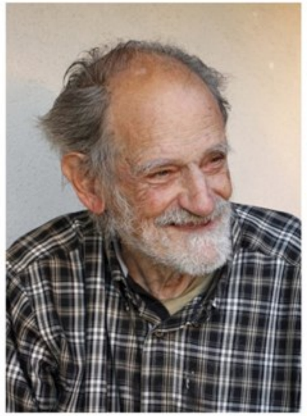
Gale-Shapely deferred acceptance algorithm

```
GALE-SHAPELY (preference lists of course instructors and TA applicants)

INITIALIZE $M$ to empty matching
WHILE ( Course $c \in C$ is free and has not offered to every applicant )
    $a \leftarrow$ Select Highest Preference $a \in$ TA of $c$ to whom $c$ has not yet offered
    IF ($a$ has not received TAship offer before)
        Add $c - a$ to matching $M$
    ELSE IF ($a$ prefers $c$ to a previously offered course by instructor $c'$)
        Replace $c' - a$ with $c - a$ in matching $M$
    ELSE
        $a$ rejects offer made by $c$

RETURN stable matching M
```


UC Berkeley


UCLA


Stanford

Stable Matching – Moving Towards *Algorithm Design*

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

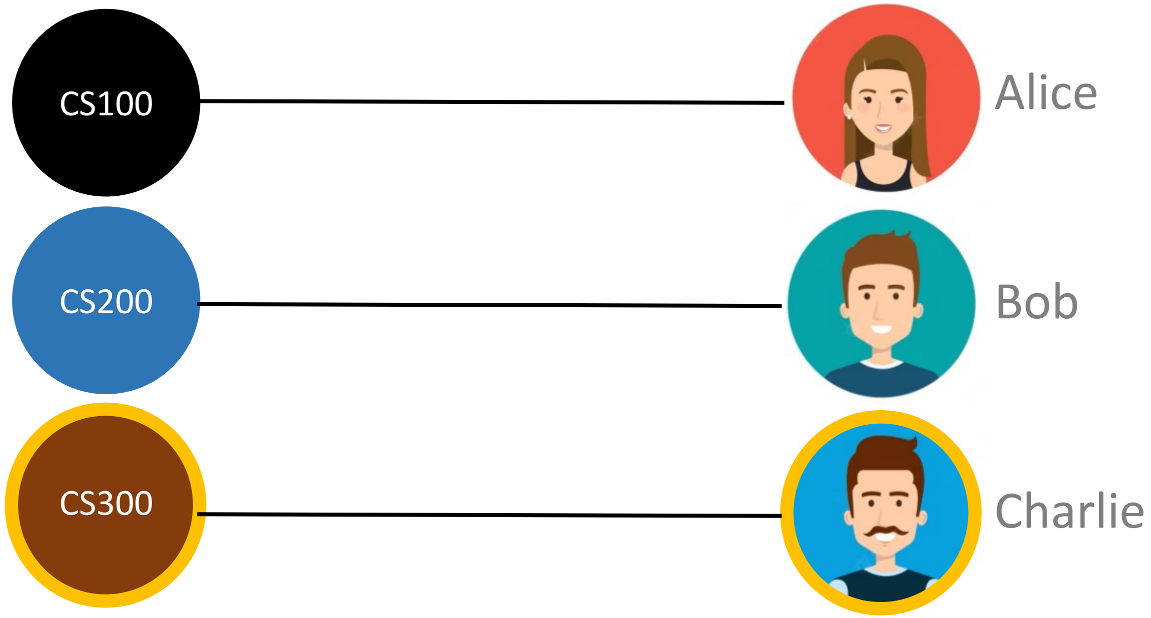
While Course *c* ∈ C is free
CS 300

Select Highest Preference *a* ∈ TA of *c to whom c has not yet proposed*
Charlie

If *a* is free then assign
(CS 300 – Charlie)

Else if *a* prefers c’ to c then c remains free

Else if *a* prefers *c* to *c’* then assign *a* to *c* and *c’* gets free



Stable Matching – Moving Towards *Algorithm Design*

7

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free

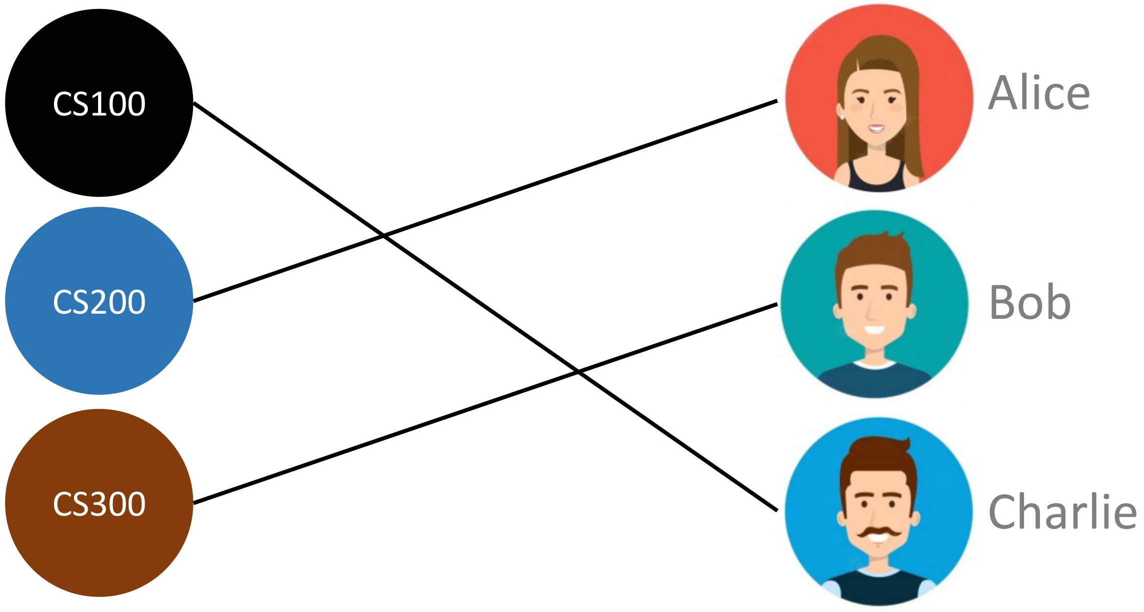
CS 100

Select Highest Preference *a* ∈ TA of *c to whom c has not yet proposed*

Charlie

If *a* is free then assign

(CS 100 - Charlie)



Else if *a* prefers *c'* to *c* then c remains free

Else if *a* prefers *c* to *c'* then assign *a* to *c* and *c'* gets free

Proof of correctness: Termination

Claim. **Algorithm terminates after** "close to" $\boldsymbol{n^2}$ **iterations of WHILE loop**

Proof.

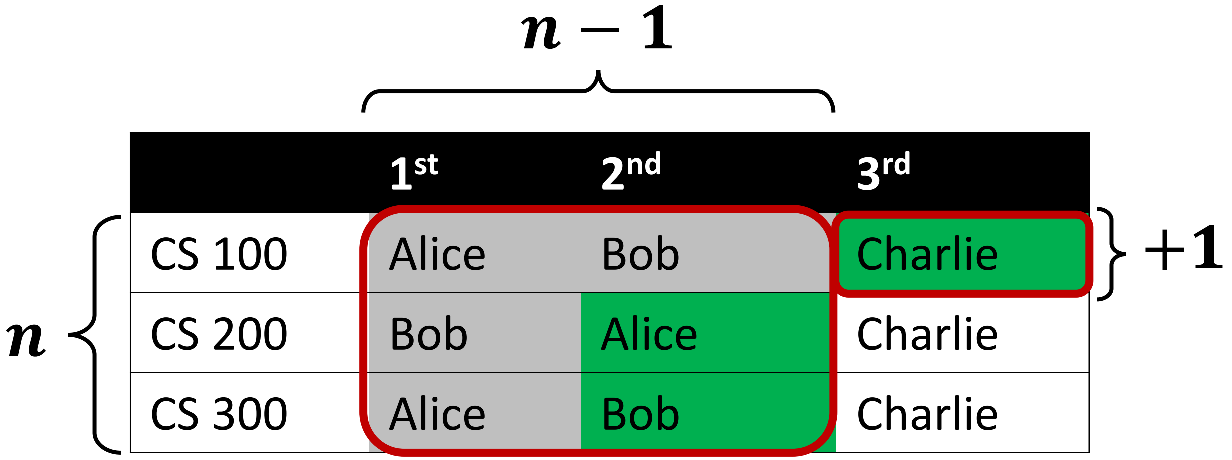
- Each time through the while loop a course instructor offers TAship to a new applicant.
- In worst case, each course instructor offer TAship to each applicant
- There can n^2possible offers at most

Formula for exact number of **worst case** iteration for Gale-Shaply algorithm is:

$$\boldsymbol{n(n-1)+1} \qquad \text{How?}$$

Proof of correctness: Termination

$$n = 3$$



$n - 1$

n

$+1$

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

$n(n-1)+1$ Why we said “close to” n^2 ?

Proof of correctness: Termination

$n(n-1)+1$ Why we said “close to” n^2 ?

$\boxed{n^2} - n + 1$

↑

Dominating Factor

n	$n^2 - n + 1$	n^2
0	1	0
500	249501	250000
1000	999001	1000000
1500	2248501	2250000
2000	3998001	4000000
2500	6247501	6250000
3000	8997001	9000000
3500	12246501	12250000
4000	15996001	16000000
4500	20245501	20250000
5000	24995001	25000000

Proof of correctness: Matching

Claim. **Gale-Shapely outputs a matching**

Proof.

- Course instructor makes an offer only if unmatched $\Rightarrow$ matched to ≤ 1 TA applicant
- TA applicant keeps only best course based on his/her preference $\Rightarrow$ matched to ≤ 1 course

Proof of correctness: Perfect Matching

Claim. **In Gale-Shapely matching, each course is assigned a TA**

Proof. [by contradiction]

- Suppose for the sake of contradiction, that some course instructor $c \in C$ is not assigned a TA upon termination of the Gale-Shapely algorithm
- Then some TA applicant, say $a \in A$, is not assigned to a course upon termination
- By Observation 2, a was never offered TAship for any course

But, c offers TAship to every applicant, since c ends up without a TA. ■

Claim. **In Gale-Shapely matching, each TA applicants is assigned to a course**

Proof. [by counting]

- By previous claim, all n courses are assigned TA (each course is matched to a TA)
- Thus, all n TA applicants get matched ■

Proof of correctness: Stability

Claim. **In Gale-Shapely matching M^*, there are no unstable pairs.**

Proof. Consider any pair $c - a$ that is not in M^*

M^*

c – a′
a – c′
. . .

- Case 1: c never made an offer to a

$\Rightarrow c$ prefers its assigned TA a' to a

$\Rightarrow c - a$ is not unstable

Course instructors make TAship offers in decreasing order of preference

- Case 2: c made an offer to a

$\Rightarrow a$ rejected c (either right away or later)

$\Rightarrow a$ prefers its assigned course c' to c

$\Rightarrow c - a$ is not unstable

TA applicants only trade up

- In either case $c - a$ is not unstable ■

Stable matching: Live Poll 2

Do all executions of Gale-Shapely lead to the same stable matching?

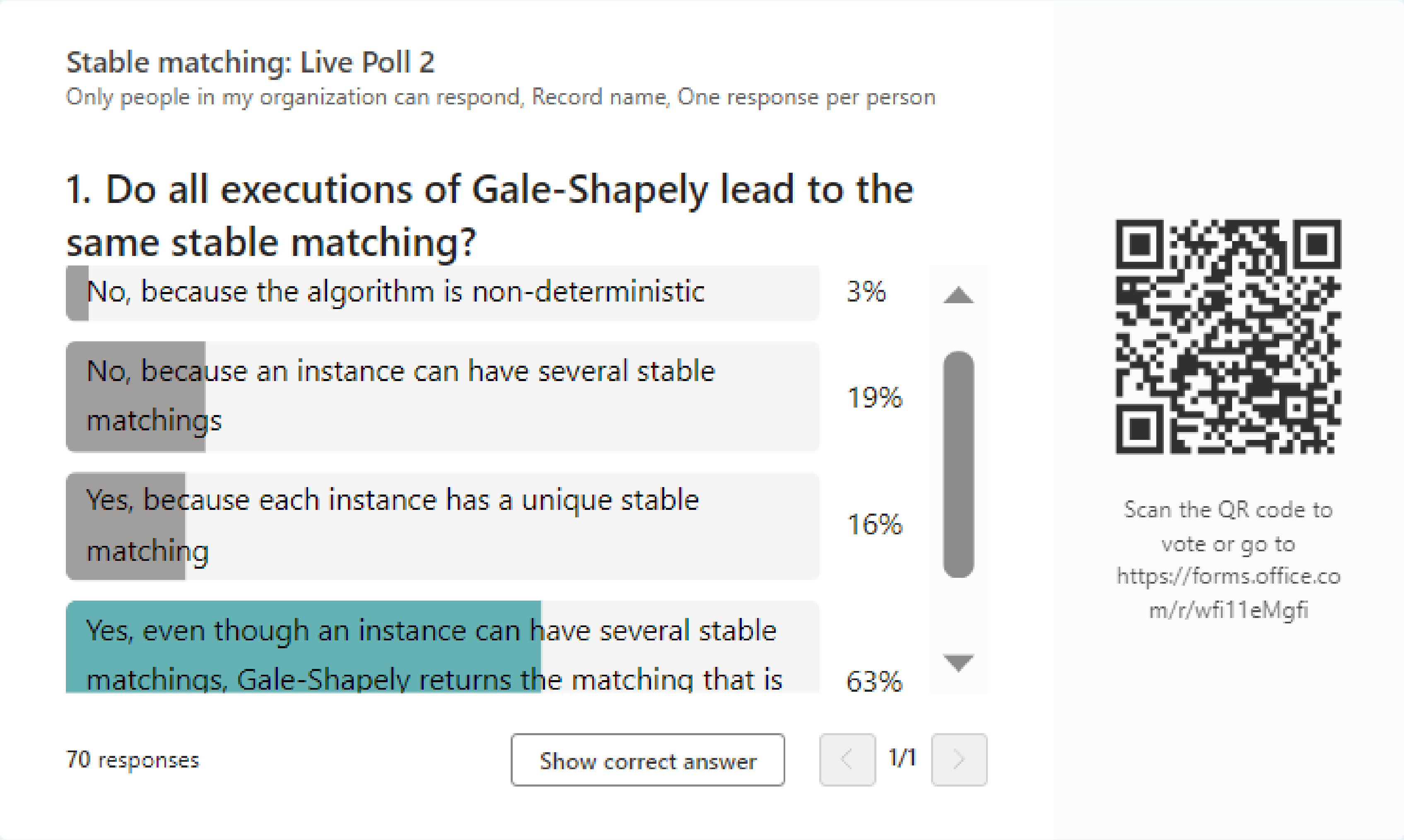
A. No, because the algorithm is non-deterministic

B. No, because an instance can have several stable matchings

C. Yes, because each instance has a unique stable matching

D. Yes, even though an instance can have several stable matchings, Gale-Shapely returns the matching that is optimal w.r.t. course instructor

Scan the QR code to vote or go to https://forms.office.com/r/wfi11eMgfi



Stable matching: Live Poll 2

Only people in my organization can respond, Record name, One response per person

1. Do all executions of Gale-Shapely lead to the same stable matching?

No, because the algorithm is non-deterministic	3%
No, because an instance can have several stable matchings	19%
Yes, because each instance has a unique stable matching	16%
Yes, even though an instance can have several stable matchings, Gale-Shapely returns the matching that is	63%

70 responses

Scan the QR code to vote or go to https://forms.office.com/r/wfi11eMgfi

Stable Matching – Moving Towards *Algorithm Design*

5

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free

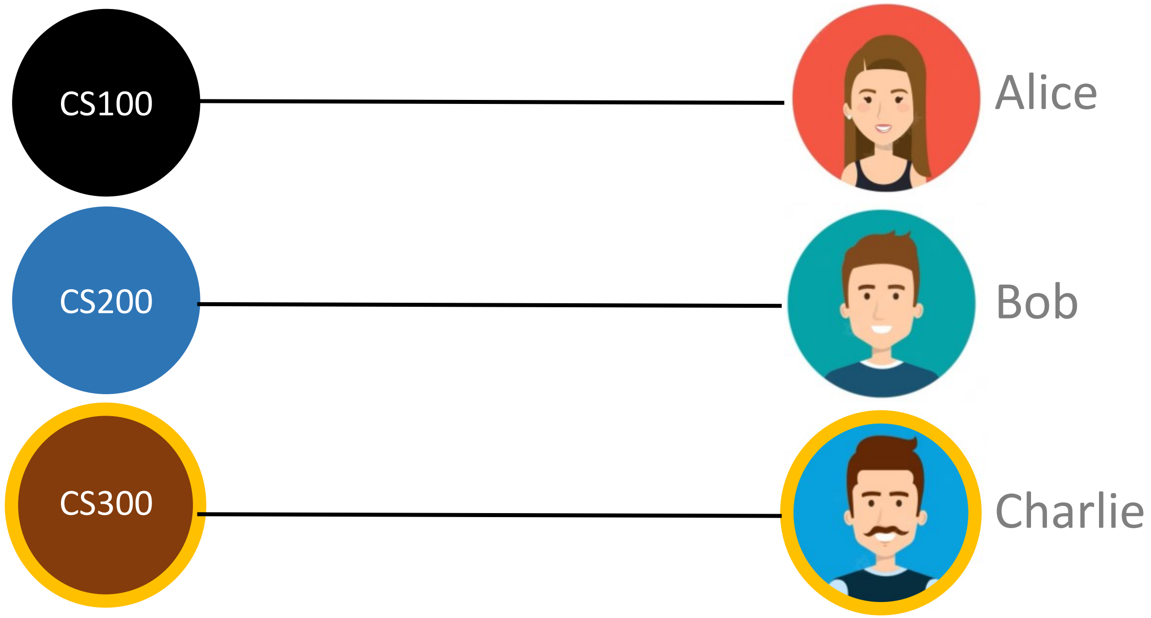
CS 300

Select Highest Preference *a* ∈ TA of *c to whom c has not yet proposed*

Charlie

If *a* is free then assign

(CS 300 – Charlie)



Else if *a* prefers c’ to c then c remains free

Else if *a* prefers *c* to *c’* then assign *a* to *c* and *c’* gets free

Stable Matching – Moving Towards ***Algorithm Design***

7

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free

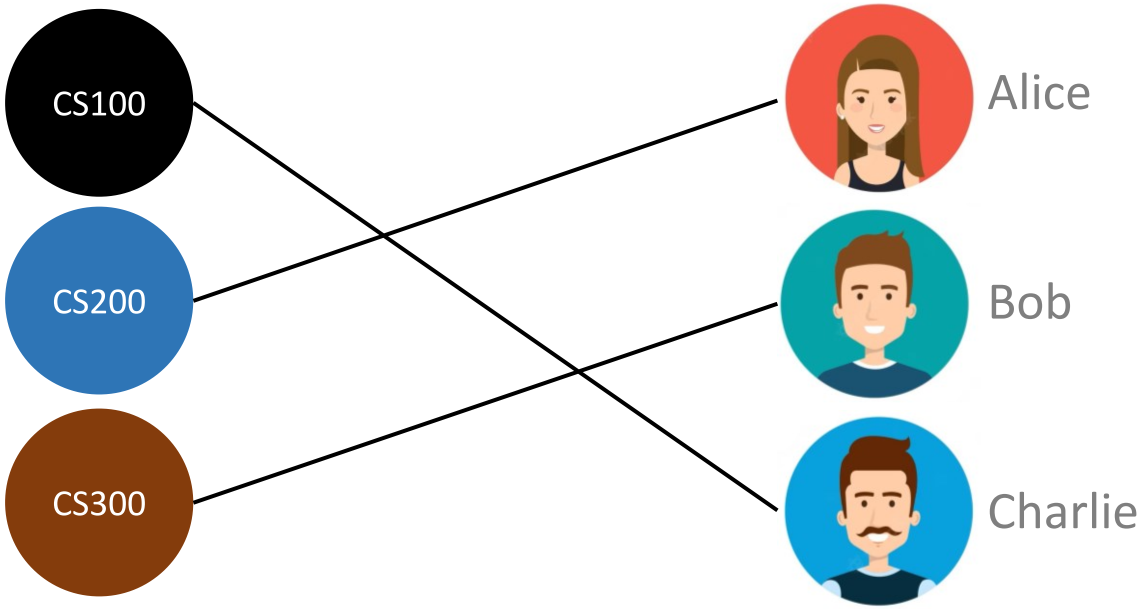
CS 100

Select Highest Preference *a* ∈ TA of *c to whom c has not yet proposed*

Charlie

If *a* is free then assign

(CS 100 - Charlie)



Else if *a* prefers *c’* to *c* then c remains free

Else if *a* prefers *c* to *c’* then assign *a* to *c* and *c’* gets free

Understanding the solution
Optimality wr.t. the course instructor preference

- For a given problem instance, there may be several stable matchings

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

Course instructors' preference list

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

TA applicants' preference list

An instance with two stable matchings

M = {(CS100 – Alice), (CS200 – Bob), (CS300 – Charlie)}

Understanding the solution
Optimality wr.t. the course instructor preference

- For a given problem instance, there may be several stable matchings

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

Course instructors' preference list

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

TA applicants' preference list

An instance with two stable matchings

M = {(CS100 – Alice), (CS200 – Bob), (CS300 – Charlie)}

M' = {(CS100 – Bob), (CS200 – Alice), (CS300 – Charlie)}

TA applicant pessimality

Q. Does Course instructor-optimality come at the expense of the TA applicants?

A. Yes

TA applicant-pessimal assignment. Each TA applicant receives the worst valid partner (not always)

Stable matching: Live Poll 3

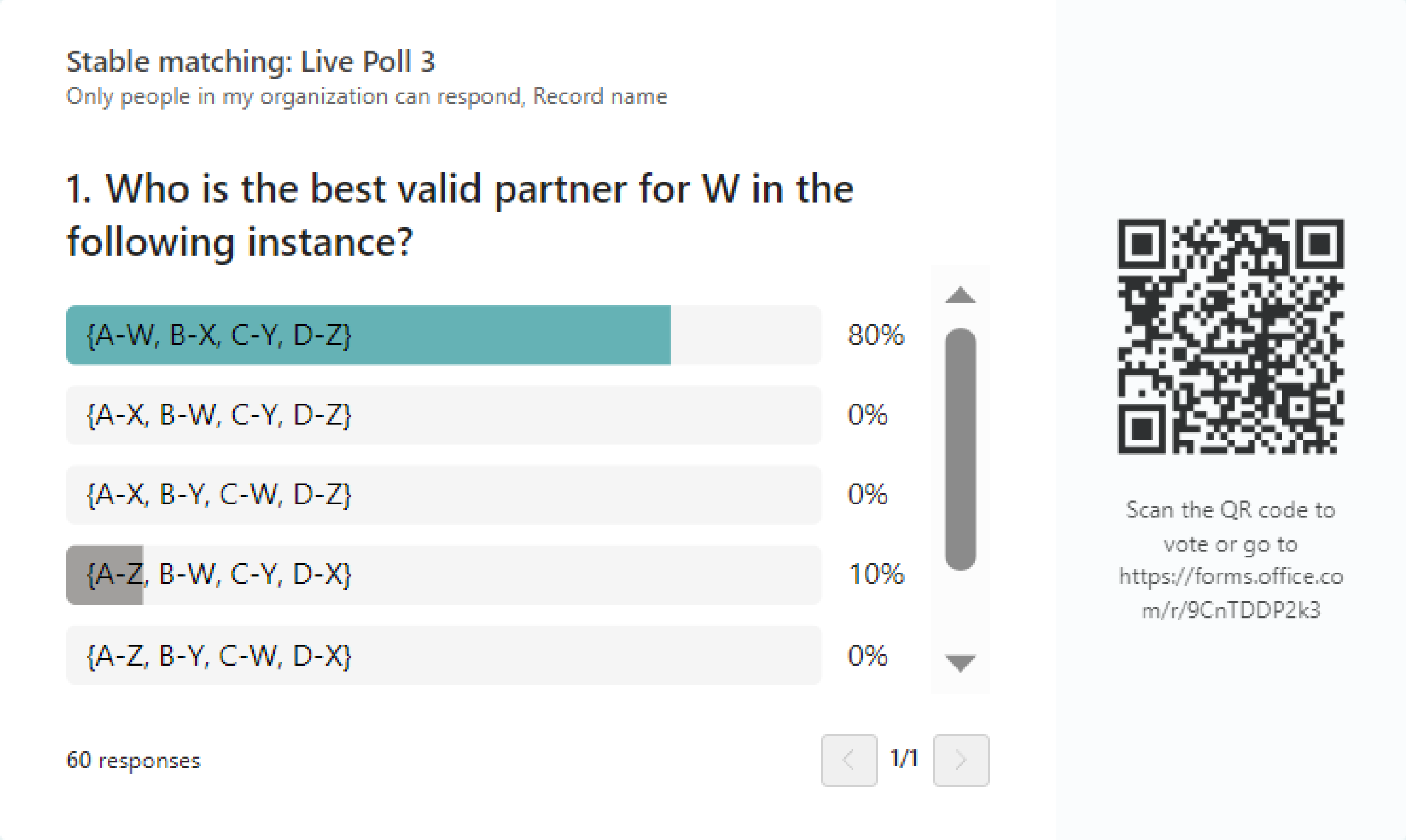
Who is the best valid partner for **W** in the following instance?

A. {A-W, B-X, C-Y, D-Z}
B. {A-X, B-W, C-Y, D-Z}
C. {A-X, B-Y, C-W, D-Z}
D. {A-Z, B-W, C-Y, D-X}
E. {A-Z, B-Y, C-W, D-X}
F. {A-Y, B-Z, C-W, D-X}
(all 6 matchings are stable)

	1st	2nd	3rd	4th
A	Y	Z	X	W
B	Z	Y	W	X
C	W	Y	X	Z
D	X	Z	W	Y

	1st	2nd	3rd	4th
W	D	A	B	C
X	C	B	A	D
Y	C	B	A	D
Z	D	A	B	C

Scan the QR code to vote or go to https://forms.office.com/r/9CnTDDP2k3



Stable matching: Live Poll 3

Only people in my organization can respond, Record name

1. Who is the best valid partner for W in the following instance?

{A-W, B-X, C-Y, D-Z}	80%
{A-X, B-W, C-Y, D-Z}	0%
{A-X, B-Y, C-W, D-Z}	0%
{A-Z, B-W, C-Y, D-X}	10%
{A-Z, B-Y, C-W, D-X}	0%

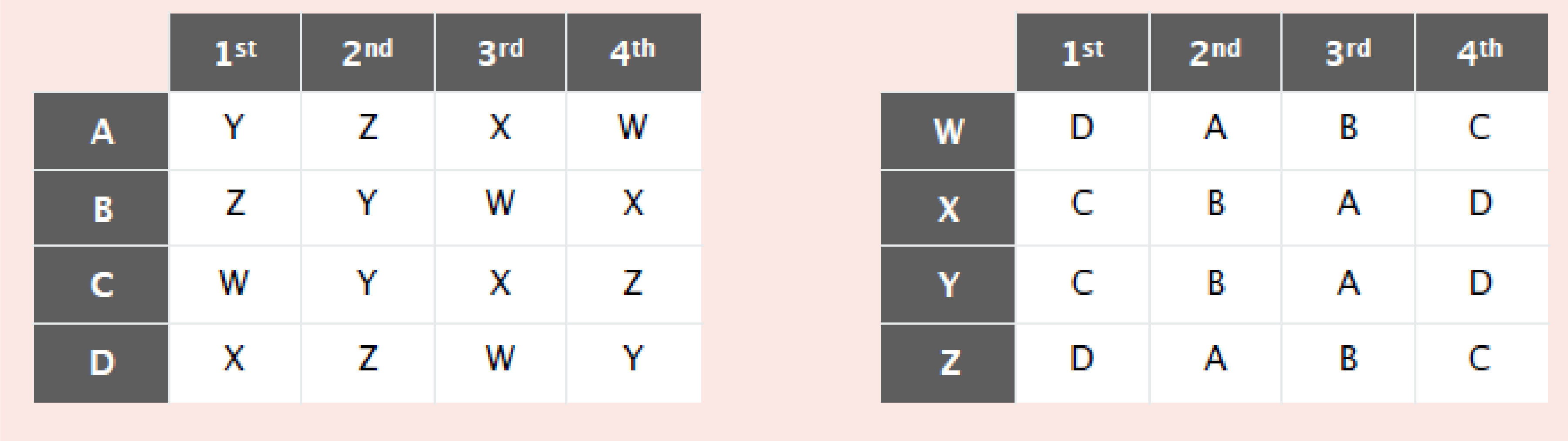
60 responses

Scan the QR code to vote or go to https://forms.office.com/r/9CnTDDP2k3

Stable matching: Live Poll 4

Which stable matchings can be found through Gale-Shapely algorithms out of these six matchings

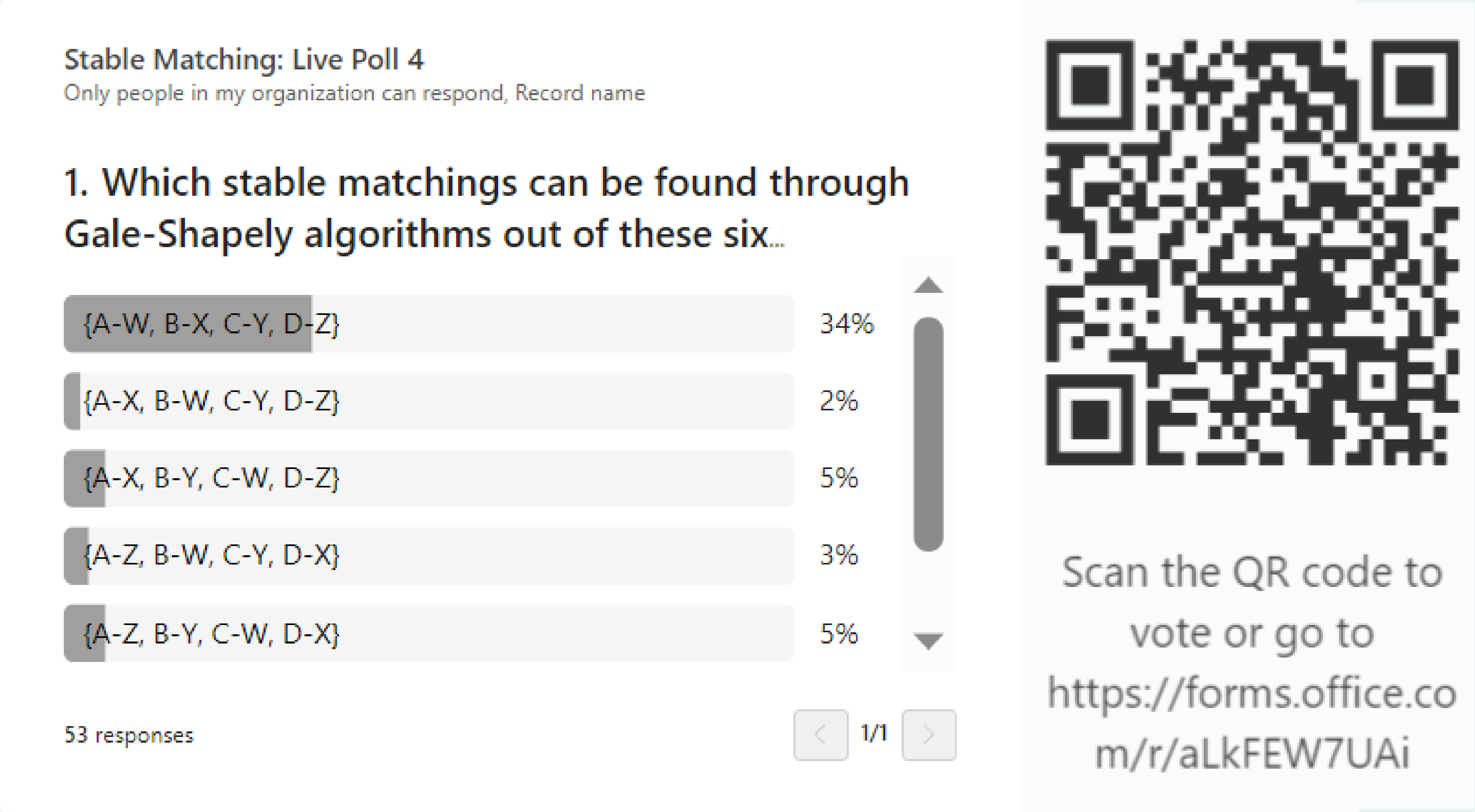
A. {A-W, B-X, C-Y, D-Z}
B. {A-X, B-W, C-Y, D-Z}
C. {A-X, B-Y, C-W, D-Z}
D. {A-Z, B-W, C-Y, D-X}
E. {A-Z, B-Y, C-W, D-X}
F. {A-Y, B-Z, C-W, D-X}



	1st	2nd	3rd	4th
A	Y	Z	X	W
B	Z	Y	W	X
C	W	Y	X	Z
D	X	Z	W	Y

	1st	2nd	3rd	4th
W	D	A	B	C
X	C	B	A	D
Y	C	B	A	D
Z	D	A	B	C

Scan the QR code to vote or go to https://forms.office.com/r/aLkFEW7UAi



Stable Matching: Live Poll 4

Only people in my organization can respond, Record name

1. Which stable matchings can be found through Gale-Shapely algorithms out of these six...

{A-W, B-X, C-Y, D-Z}	34%
{A-X, B-W, C-Y, D-Z}	2%
{A-X, B-Y, C-W, D-Z}	5%
{A-Z, B-W, C-Y, D-X}	3%
{A-Z, B-Y, C-W, D-X}	5%

53 responses

Scan the QR code to vote or go to https://forms.office.com/r/aLkFEW7UAi

Summary

Stable matching problem. Given n course instructors and n TA applicants, and their preference lists, find a stable matching if one exists.

Theorem. [Gale-Shapely 1962] The Gale-Shapely algorithm guarantees to find a stable matching for **any** problem instance.

Stable roommate problem

Q. Do stable matching always exist?

A. Not always

Stable roommate problem.

- n people; each person ranks others from 1 to $n-1$
- Assign roommate pairs so that no unstable pairs

	1st	2nd	3rd
A	B	C	D
B	C	A	D
C	A	B	D
D	A	B	C

No perfect matching is stable

$A - B, C - D \quad \Rightarrow \quad B - C$ unstable

$A - C, B - D \quad \Rightarrow \quad B - A$ unstable

$A - D, B - C \quad \Rightarrow \quad A - C$ unstable

Observation. Stable matching not always exist

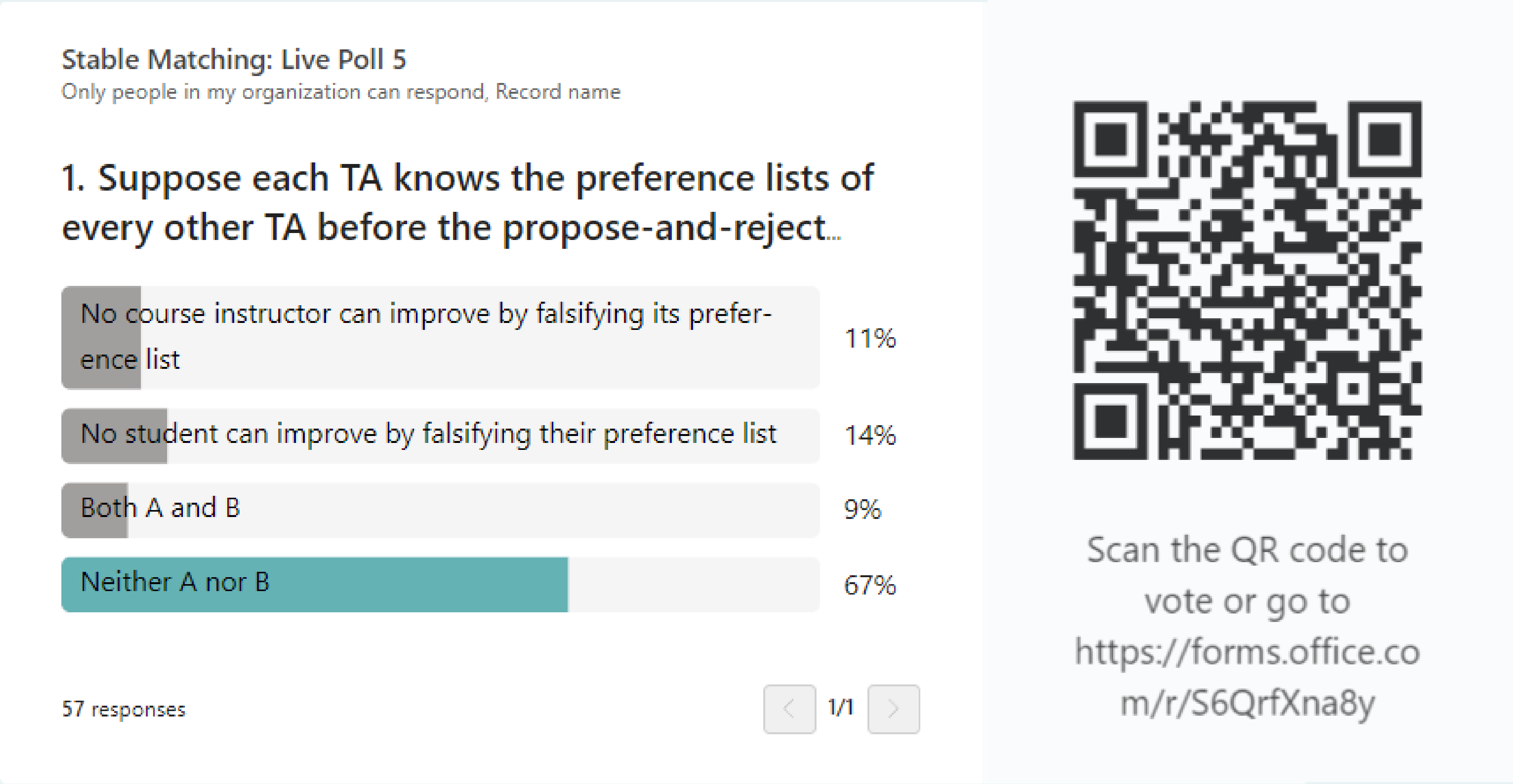
Stable matching: Live Poll 5

Lie for gain?

Suppose each **TA** knows the preference lists of every other **TA** before the propose-and-reject algorithm is executed. Which of the following is true?

A. No, course instructor can improve by falsifying its preference list
B. No, student can improve by falsifying their preference list
C. Both A and B
D. Neither A nor B

Scan the QR code to vote or go to https://forms.office.com/r/S6QrfXna8y



Stable Matching: Live Poll 5

Only people in my organization can respond, Record name

1. Suppose each TA knows the preference lists of every other TA before the propose-and-reject…

No course instructor can improve by falsifying its preference list	11%
No student can improve by falsifying their preference list	14%
Both A and B	9%
Neither A nor B	67%

57 responses

Scan the QR code to vote or go to https://forms.office.com/r/S6QrfXna8y

Deceit of Gale-Shapley Algorithm

Original

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Evil Alice

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Deceit of Gale-Shapley Algorithm

Original

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Evil Alice

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Deceit of Gale-Shapley Algorithm

Original

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Evil Alice

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Deceit of Gale-Shapley Algorithm

Original

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Evil Alice

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Deceit of Gale-Shapley Algorithm

Original

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Evil Alice

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Deceit of Gale-Shapley Algorithm

Original

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Evil Alice

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Deceit of Gale-Shapley Algorithm

Original

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Evil Alice

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Now

	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

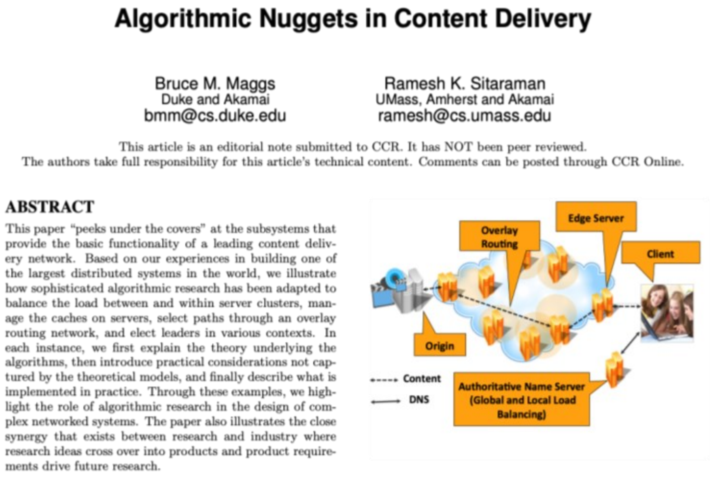
Stable Matching Problem

- Selecting teaching assistants (TA) for courses – TA-course matching problem
- Matching medical students to hospitals
- Matching employers to applicants for job hiring
- College admission – matching students to colleges
- Content delivery networks – assigning users to web servers

A modern application

Content delivery networks. Distribute much of world's content on web.

- **User.** Preferences based on latency and packet loss.
- **Web server.** Preferences based on costs of bandwidth and co-location.
- **Goal.** Assign billions of users to 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.

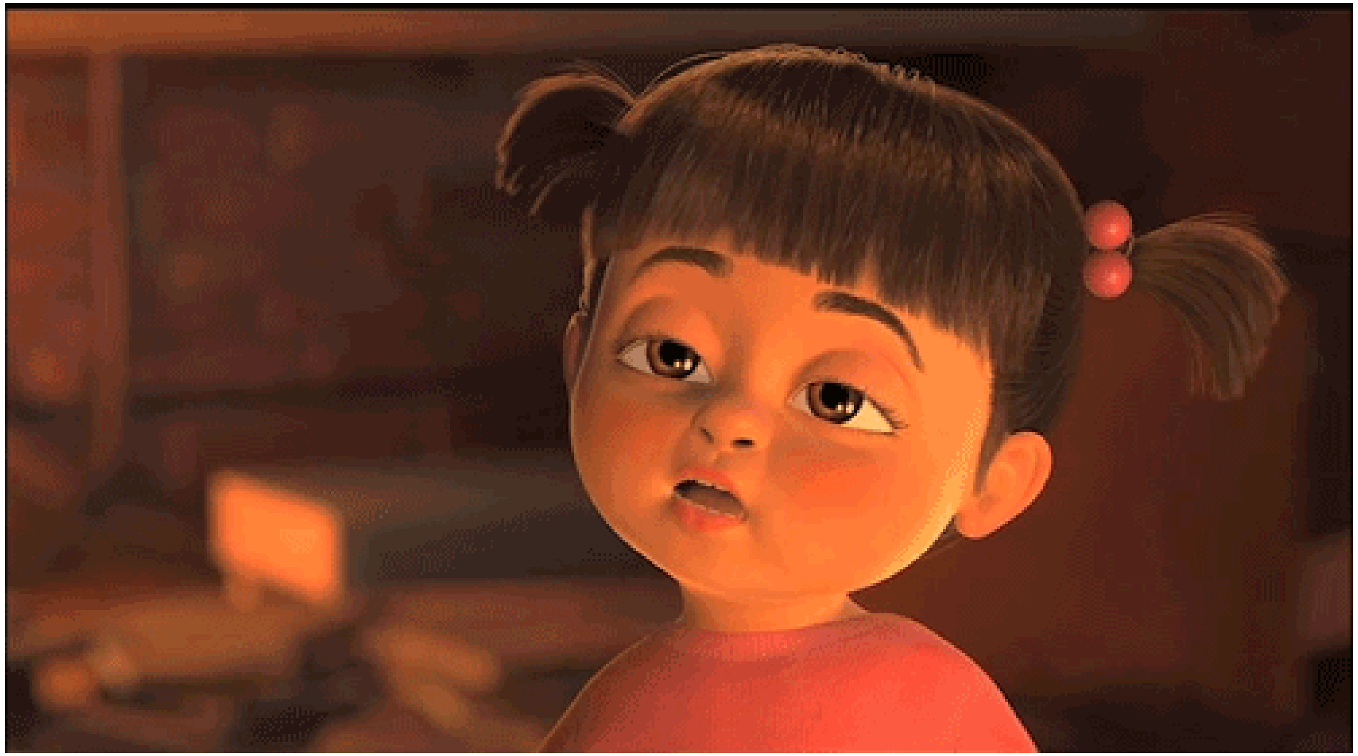
ABSTRACT

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.

Slide credit: Kevin Wayne. Theory of Algorithms (COS423)

www.cs.princeton.edu/courses/archive/spring18/cos423

Thanks a lot



If you are taking a Nap, **wake up**........Lecture Over