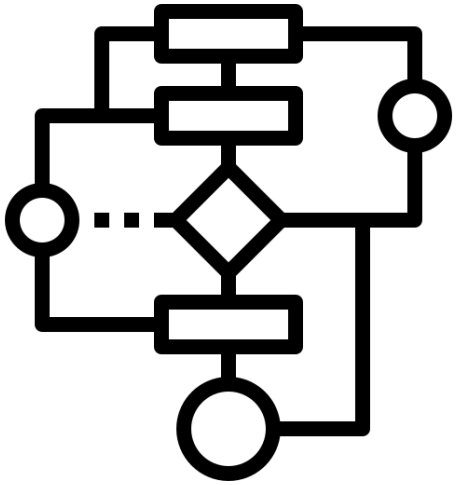
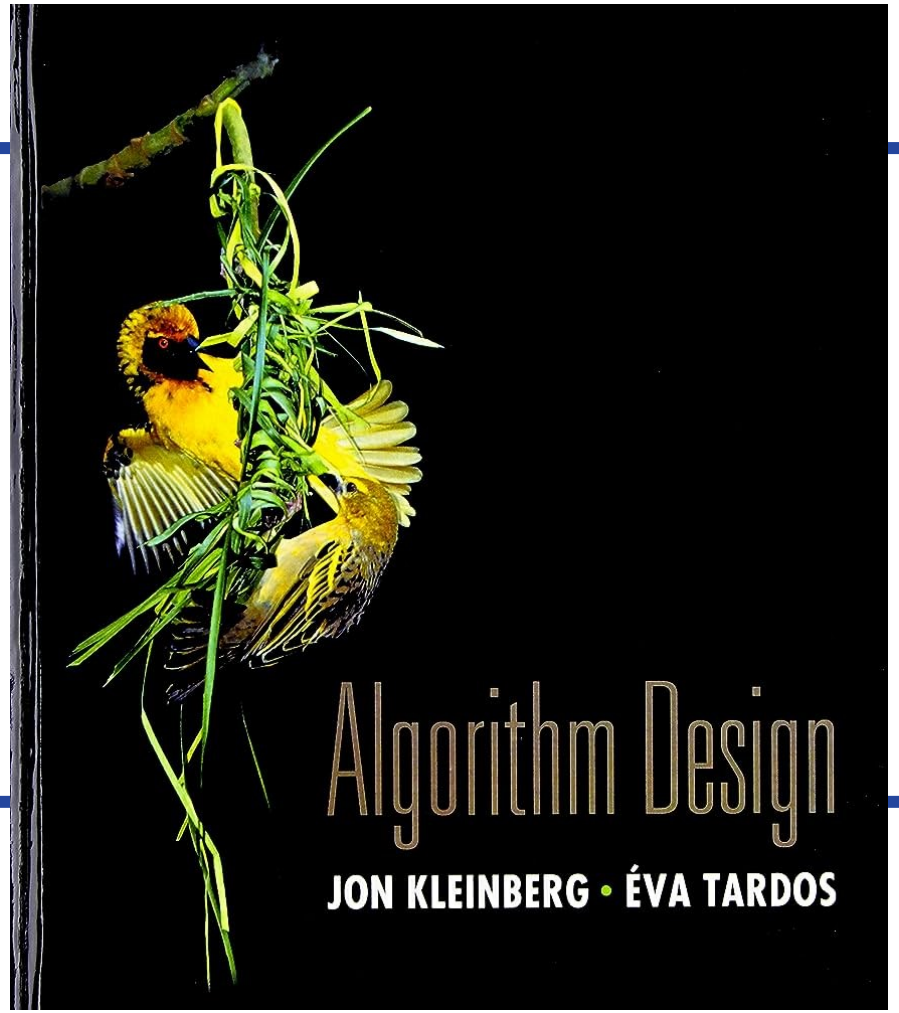


CS 310: Algorithms

Lecture 2

Instructor: Naveed Anwar Bhatti



Chapter 1:
Introduction and Some Representative Problems

Unstable pair

- Def. Given a perfect matching M, course instructor $\boldsymbol{c}$ and a TA applicant $\boldsymbol{a}$ form an unstable pair if both
  - c prefers a to the assigned TA
  - a prefers c to the assigned course

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

	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

(CS100 – Bob) and **(CS100 – Alice)** are unstable pair

Unstable pair – Through Bipartite Graph

- Def. Given a perfect matching M, course instructor c and a TA applicant a form an unstable pair if both
  - c prefers a to the assigned TA
  - a prefers c to the assigned course

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

	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

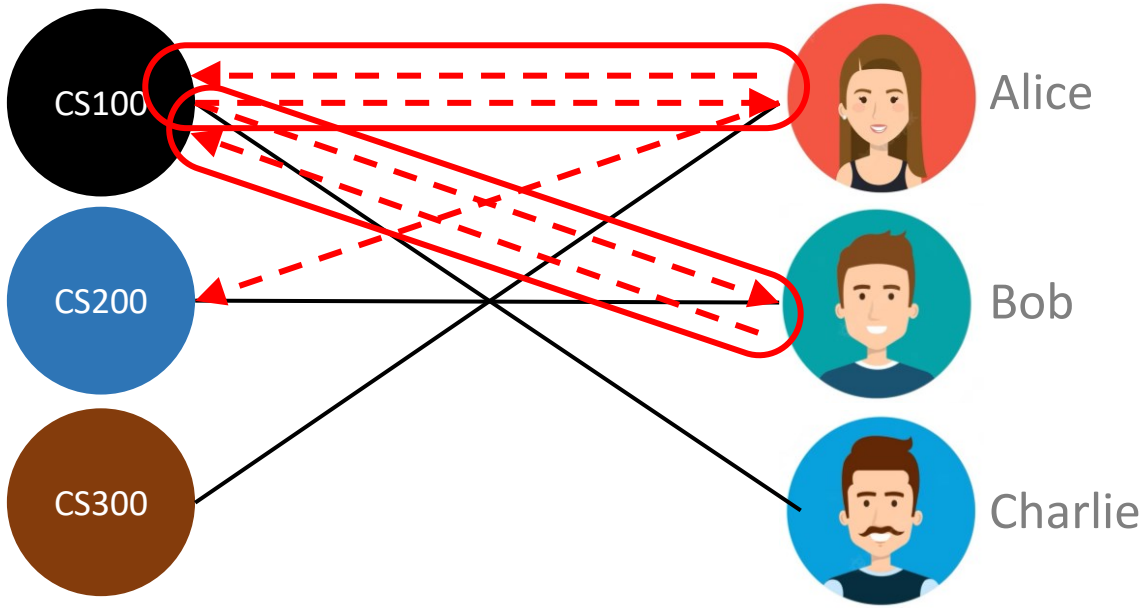
(CS100 – Bob) and **(CS100 – Alice)** are unstable pair

Unstable pair – Through Bipartite Graph

	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

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



(CS100 – Bob) and **(CS100 – Alice)** are unstable pair

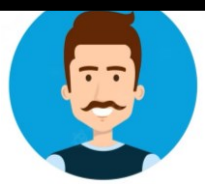
Stable Matching – Moving Towards ***Algorithm Design***

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

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

But first...

CS300



Charlie

More Administrivia…

We can have **Surprise Quizzes**

*if attendance drops below **50%**

Assignments will be **individuals**

Assignments will be **non-coding** based

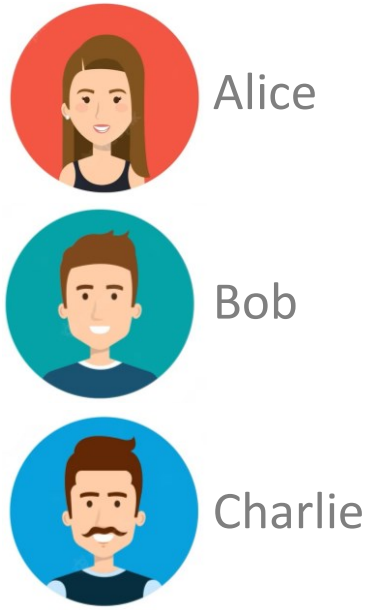
TAs almost finalized (*Perfect and Stable-Matching found*)

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

Select Course c ∈ C
CS 100



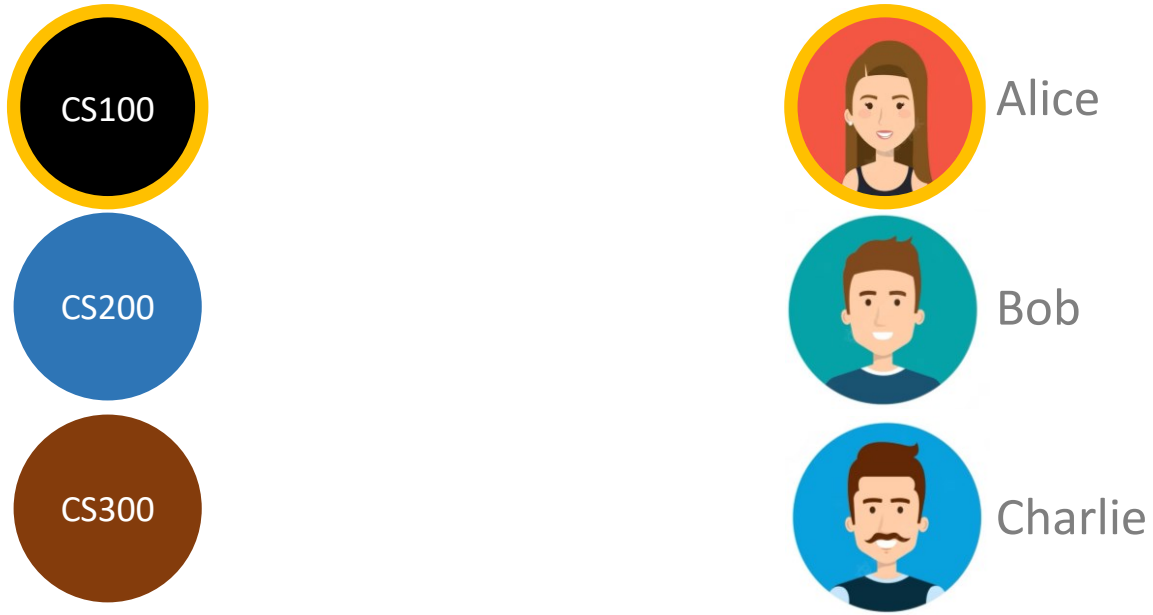
Stable Matching – Moving Towards *Algorithm Design*

	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

Select Course $c \in C$
CS 100

Select Highest Preference $a \in$ TA of c
Alice



Stable Matching – Moving Towards *Algorithm Design*

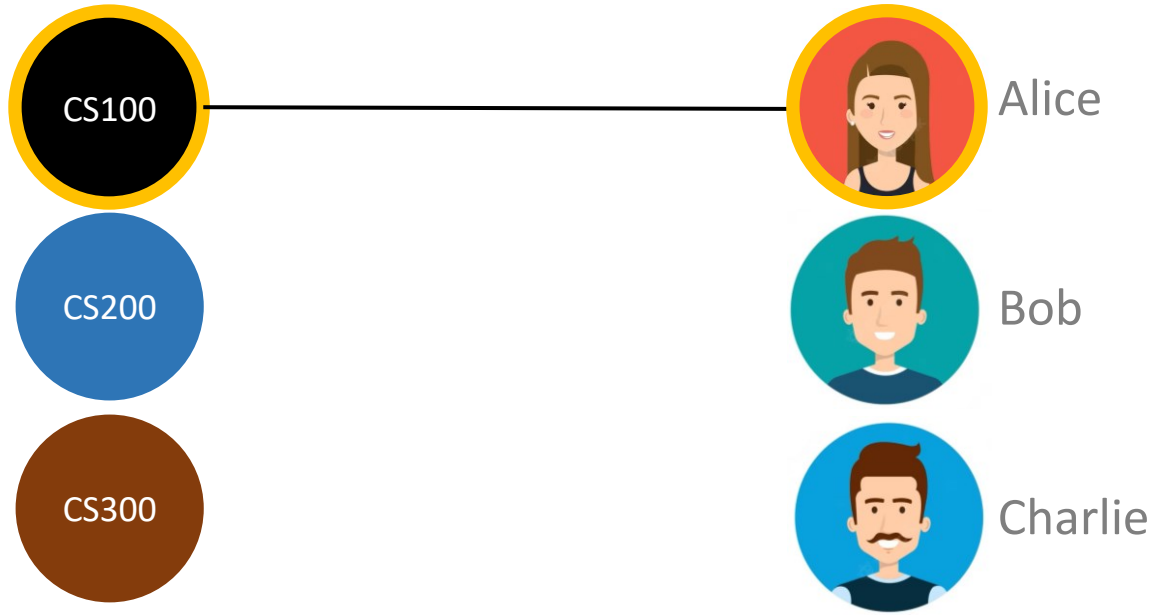
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

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

Select Course $c \in C$
CS 100

Select Highest Preference $a \in$ TA of c
Alice

If a is free then assign
(CS 100 – Alice)



Stable Matching – Moving Towards *Algorithm Design*

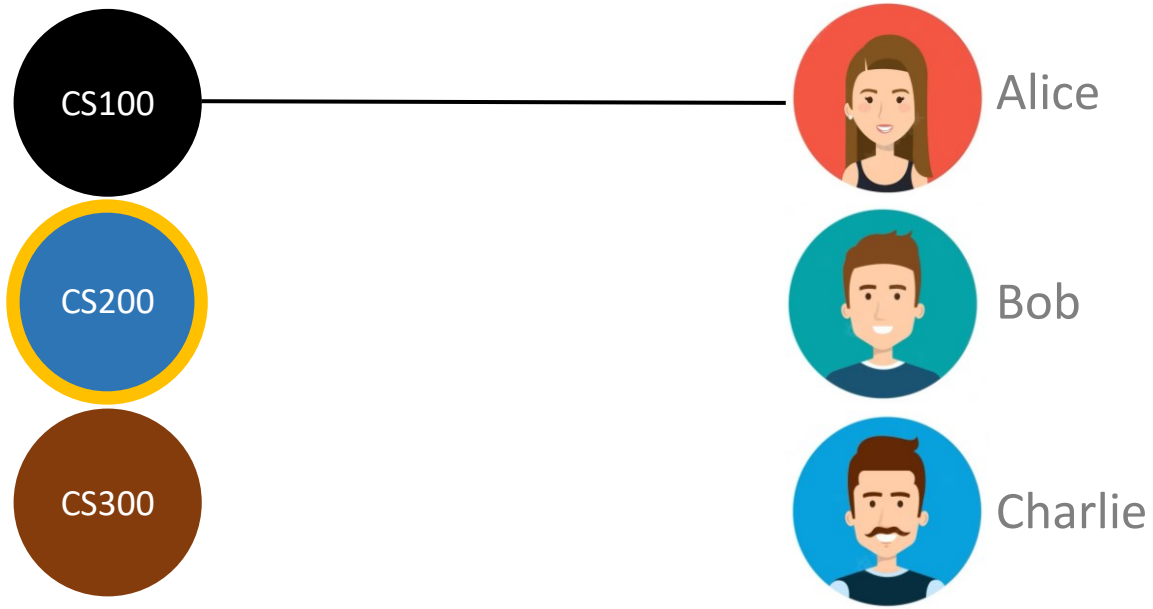
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

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

Select Course $c \in C$
CS 200

Select Highest Preference $a \in$ TA of c

If a is free then assign



Stable Matching – Moving Towards *Algorithm Design*

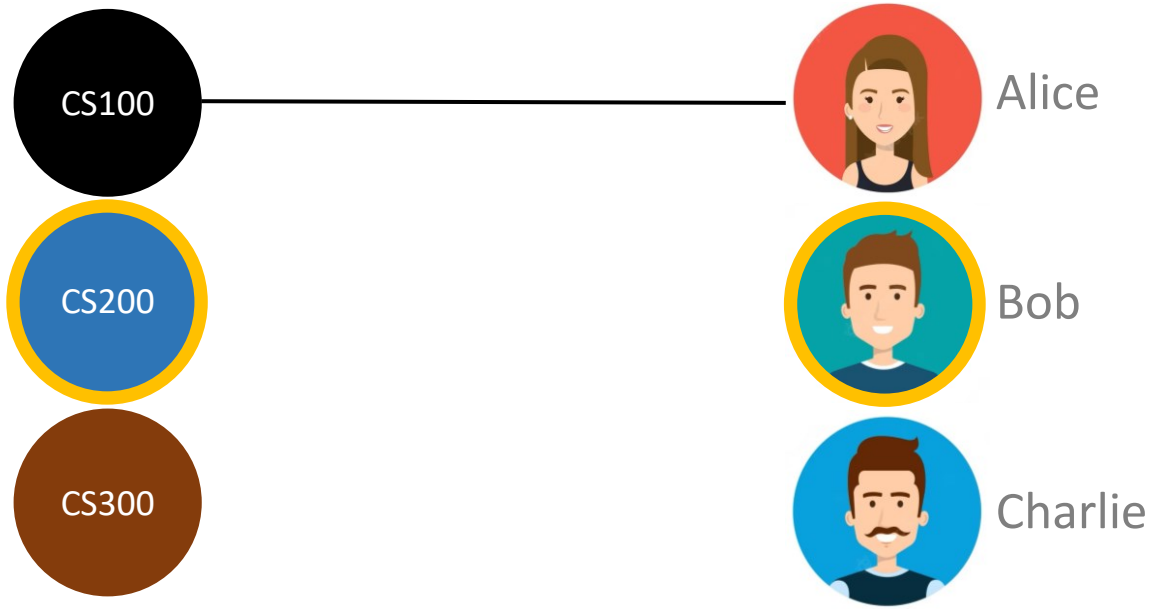
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

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

Select Course $c \in C$
CS 200

Select Highest Preference $a \in$ TA of c
Bob

If a is free then assign



Stable Matching – Moving Towards *Algorithm Design*

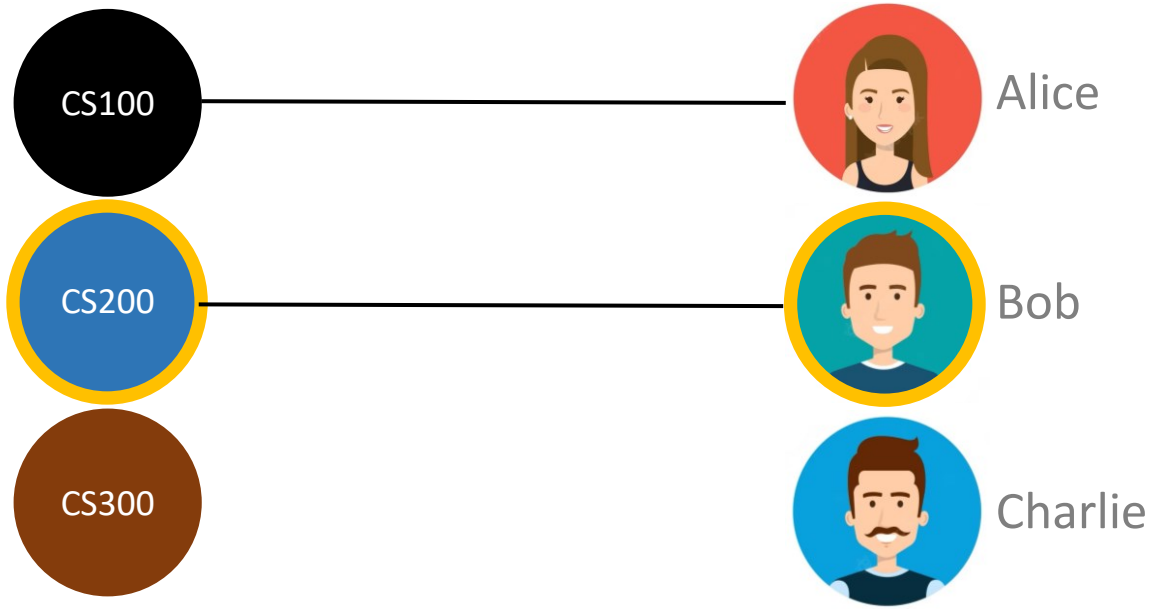
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

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

Select Course $c \in C$
CS 200

Select Highest Preference $a \in$ TA of c
Bob

If a is free then assign
(CS 200 – Bob)



Stable Matching – Moving Towards *Algorithm Design*

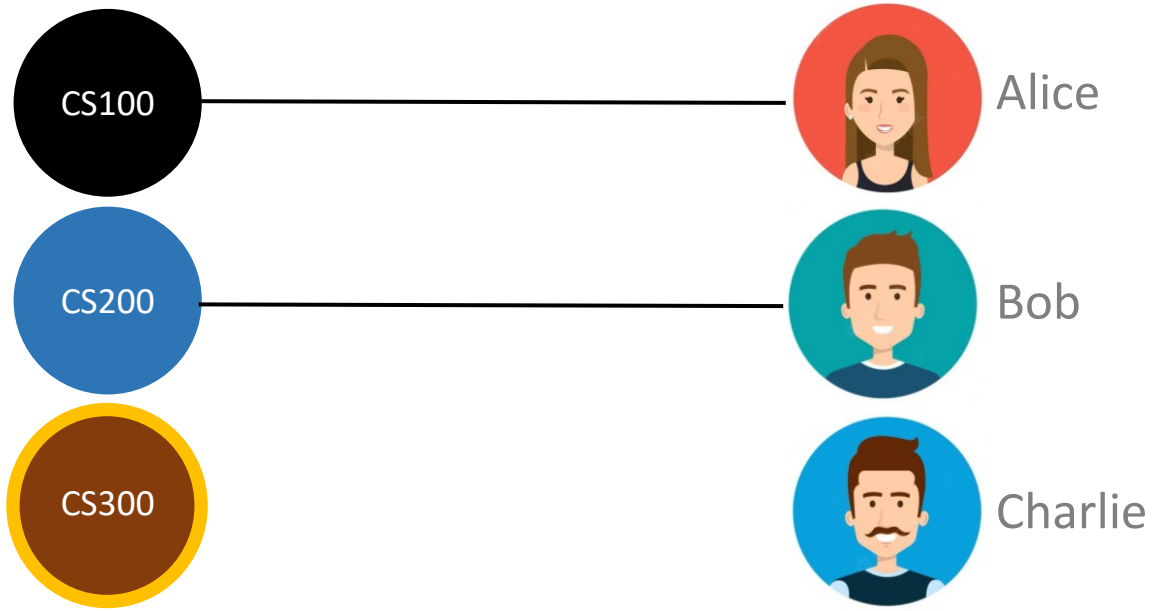
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

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

Select Course $c \in C$
CS 300

Select Highest Preference $a \in$ TA of c

If a is free then assign



Stable Matching – Moving Towards *Algorithm Design*

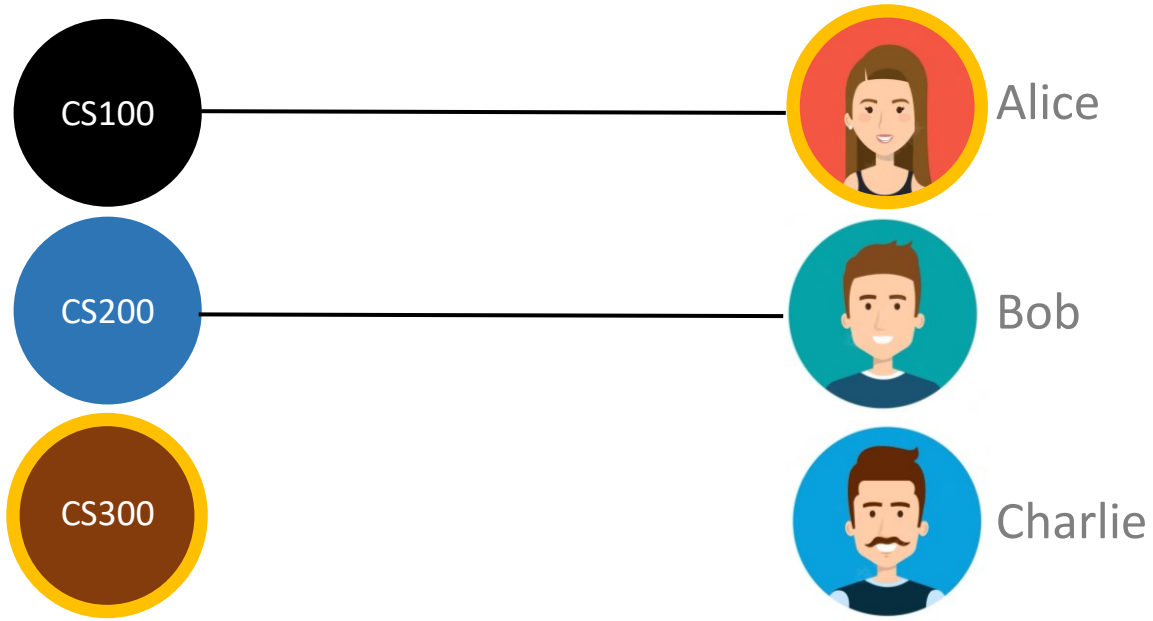
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

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

Select Course $c \in C$
CS 300

Select Highest Preference $a \in$ TA of c
Alice

If a is free then assign



Stable Matching – Moving Towards *Algorithm Design*

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

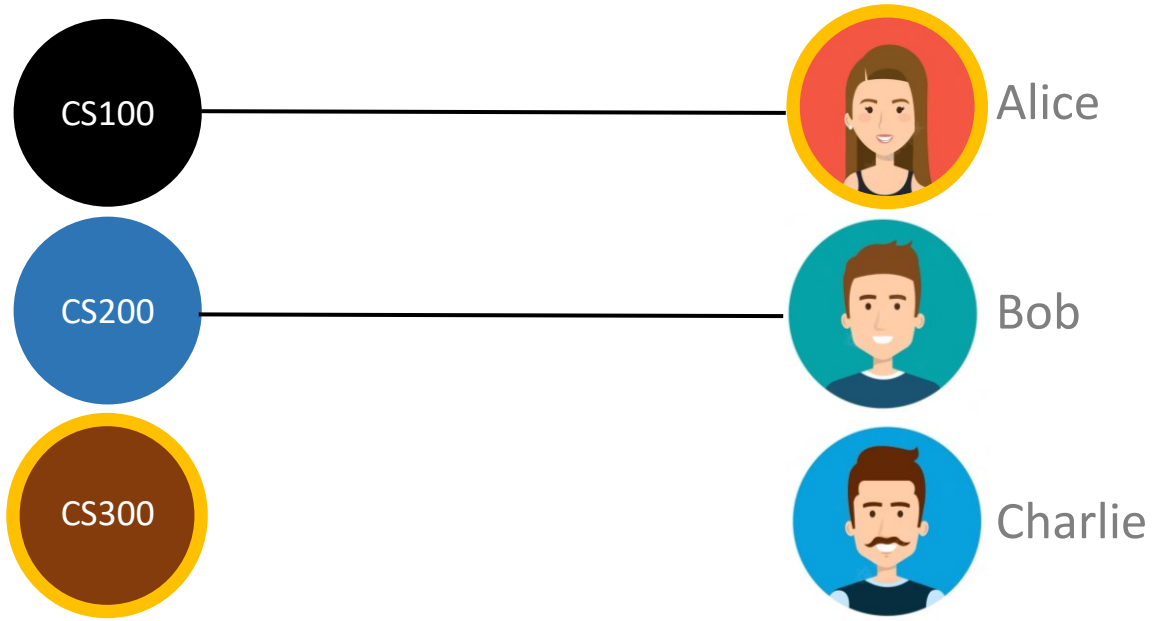
	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

Select Course $c \in C$
CS 300

Select Highest Preference $a \in$ TA of c
Alice

If a is free then assign
Already assigned

We need to add additional condition



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

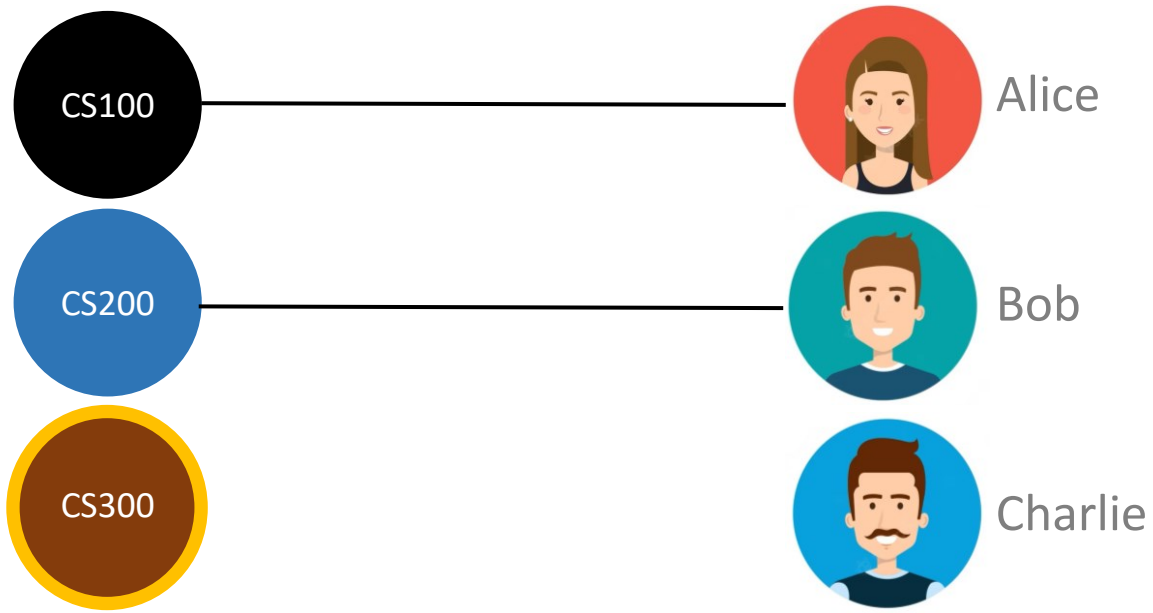
Now what? We need to clarify this condition

Select Course *c* ∈ C
CS 300

Select Highest Preference *a* ∈ TA of *c*
Alice

If *a* is free then assign
Already assigned

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



Stable Matching – Moving Towards *Algorithm Design*

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

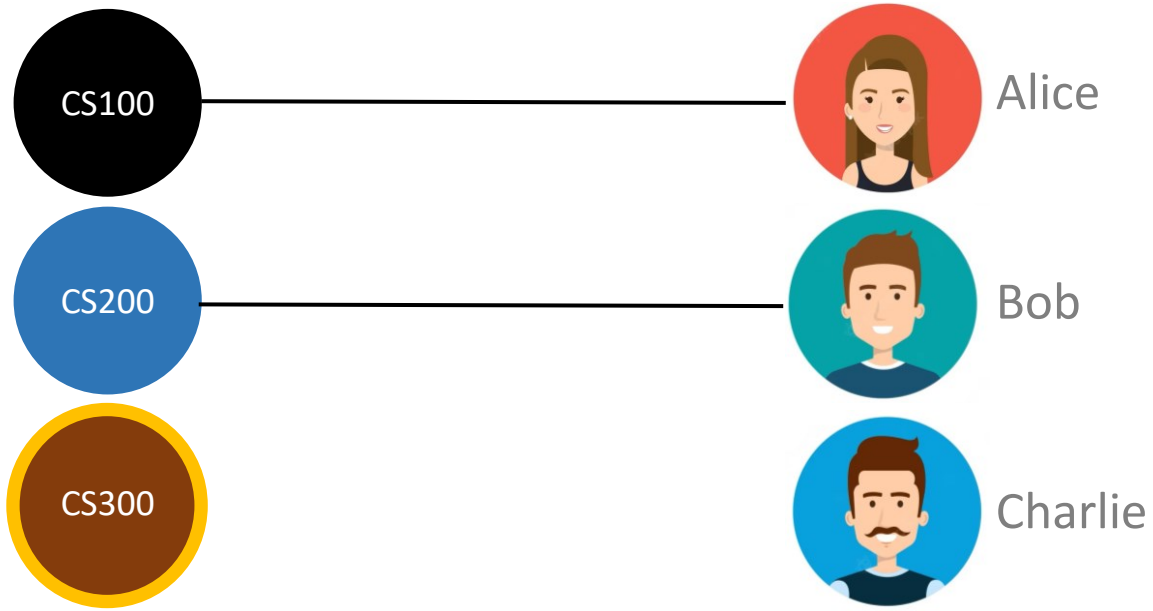
	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 300

Select Highest Preference *a* ∈ TA of *c*

If *a* is free then assign

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



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

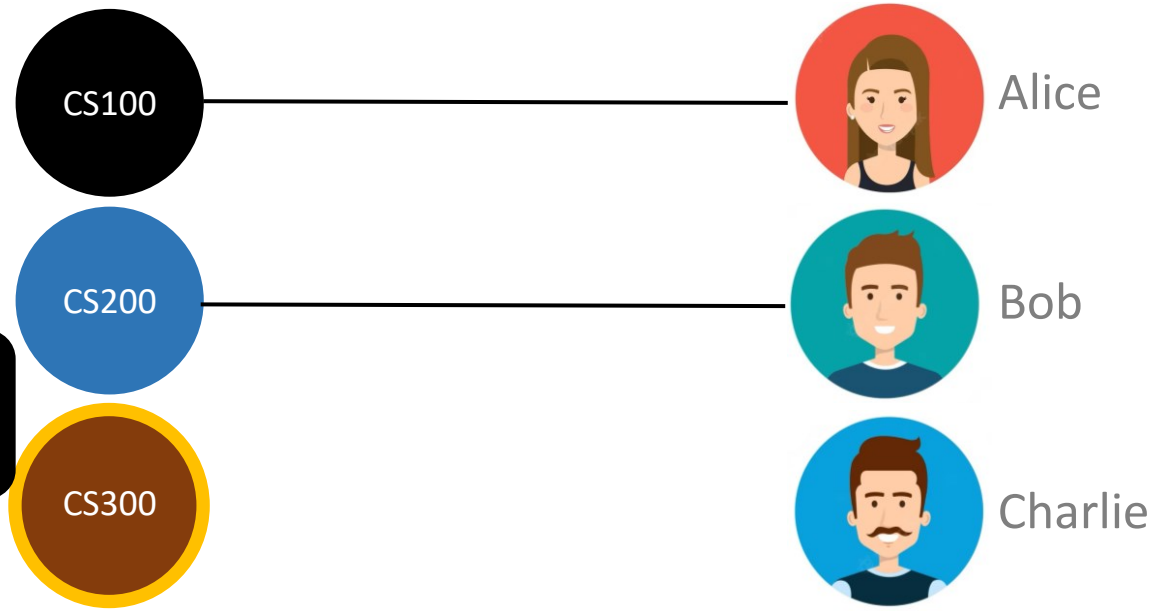
While Course *c* ∈ C is free
CS 300

Select Highest Preference *a* ∈ TA of *c*
Alice

If *a* is free then assign
Already assigned

Stuck in loop. We need to modify this condition

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



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

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

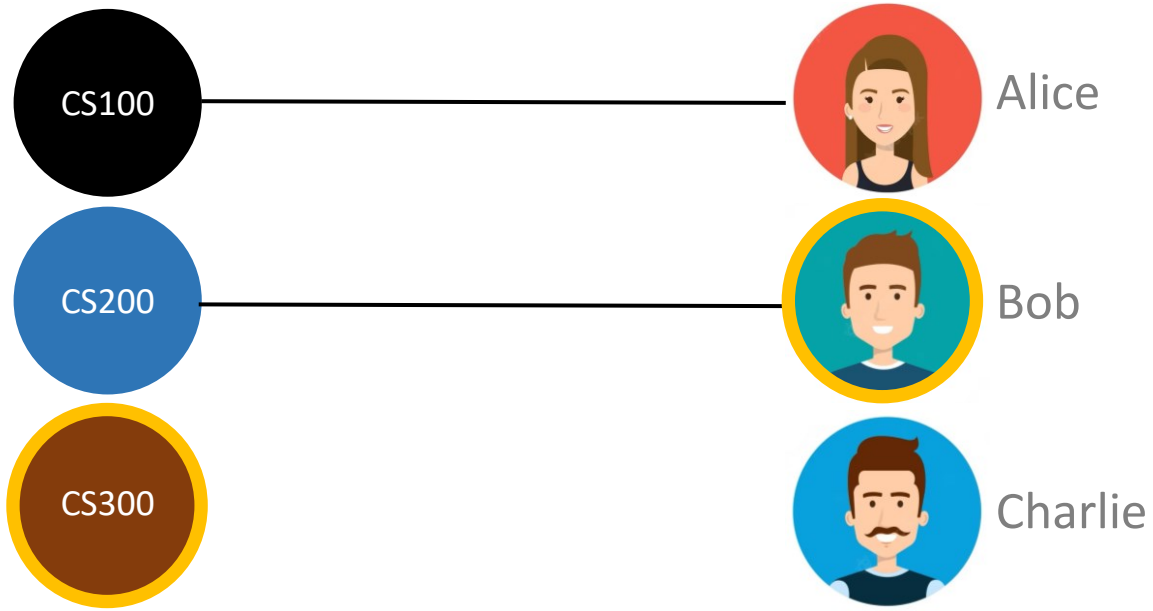
	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 300

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

If *a* is free then assign

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



Stable Matching – Moving Towards *Algorithm Design*

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

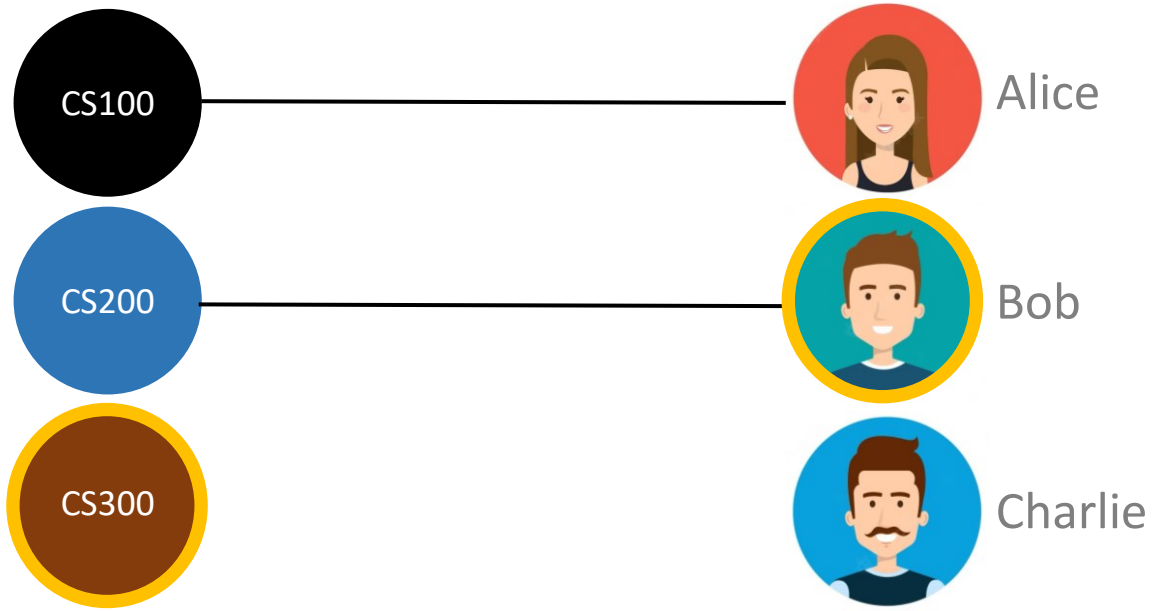
	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 300

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

If *a* is free then assign
Already assigned

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



Stable Matching – Moving Towards *Algorithm Design*

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

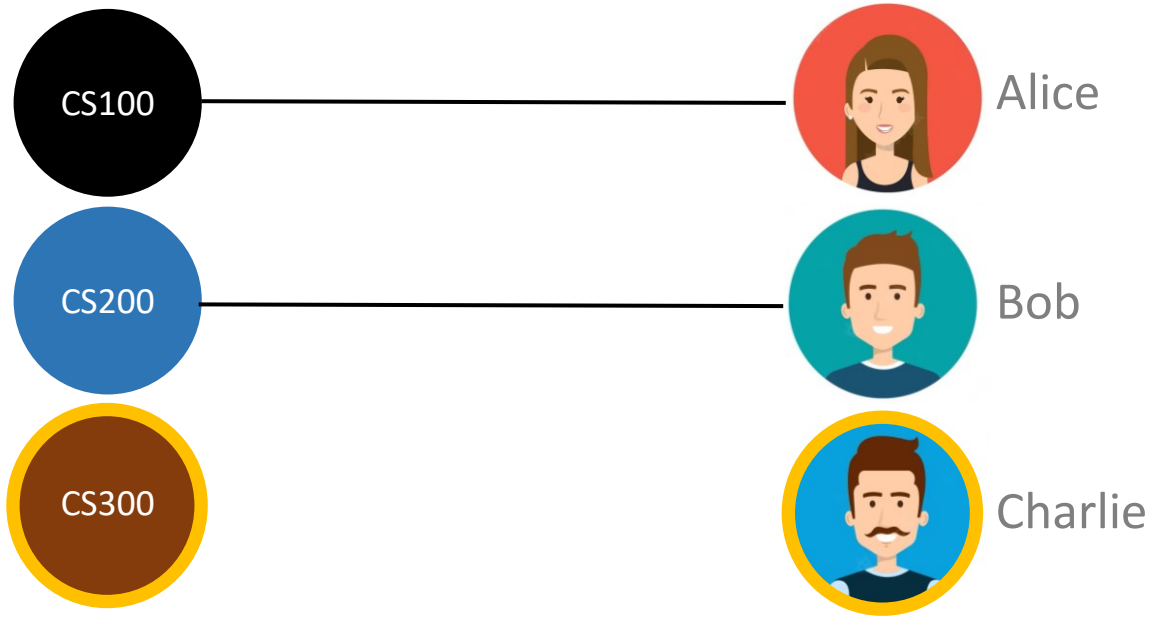
	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

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

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



Stable Matching – Moving Towards *Algorithm Design*

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

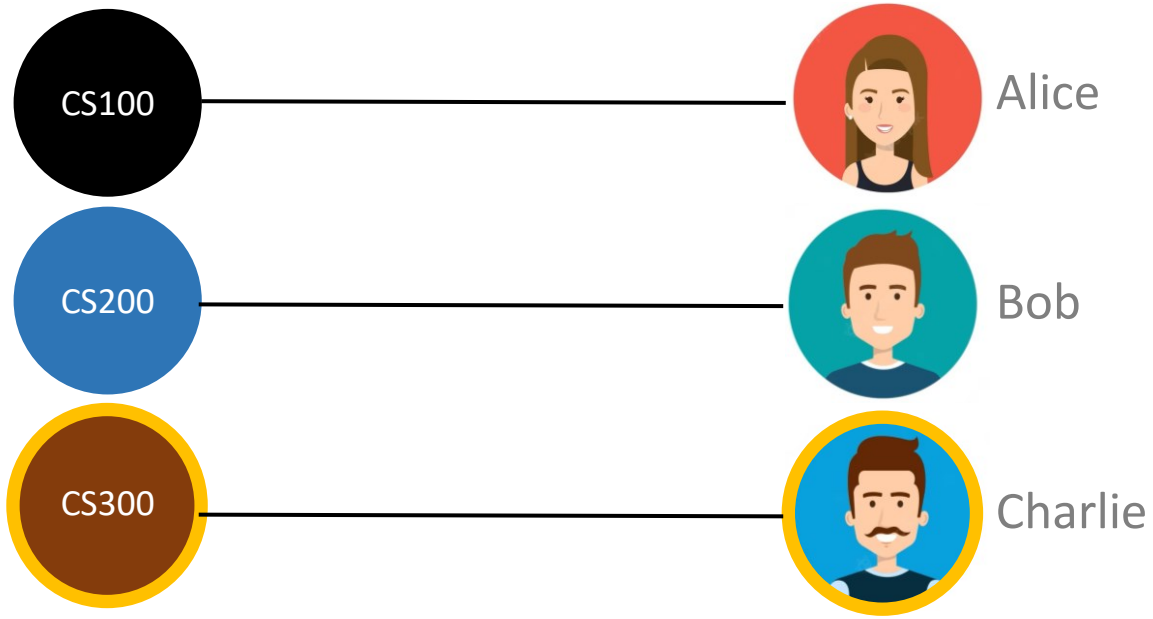
	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

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



Stable Matching – Moving Towards *Algorithm Design*

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

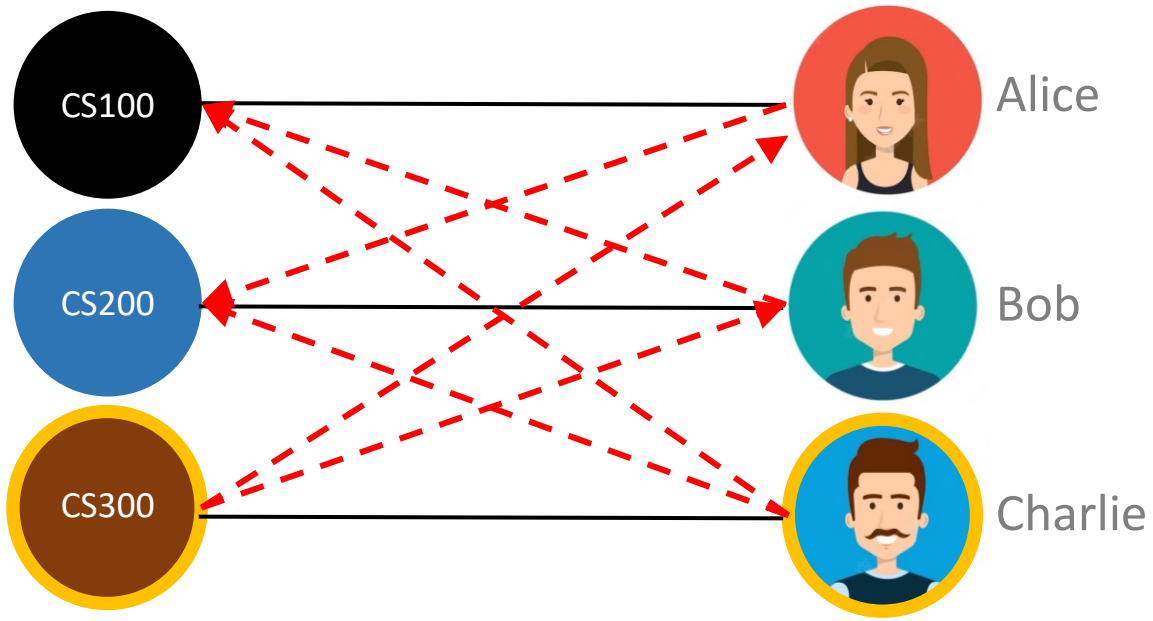
	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

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



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

Stable Matching – Moving Towards *Algorithm Design*

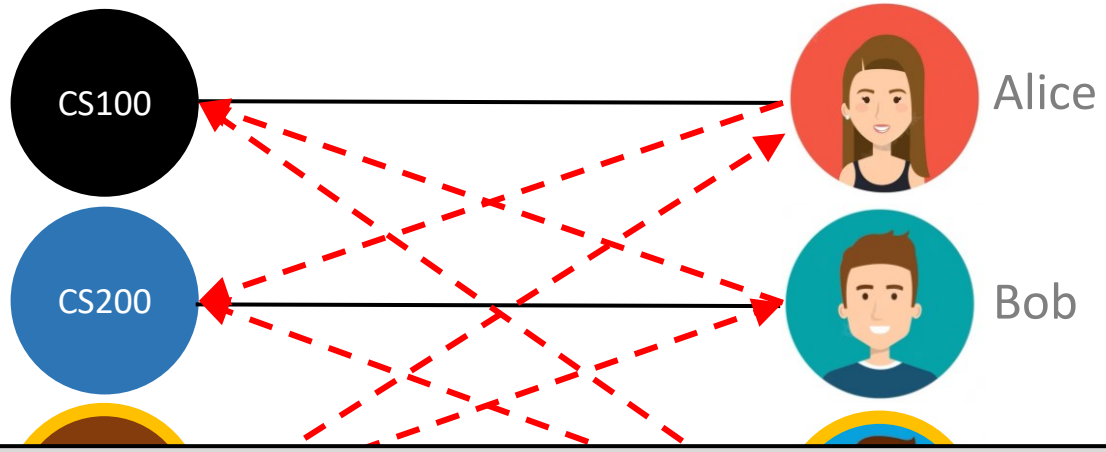
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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



Observation 1. Course instructors make offer to TA applicants in decreasing order of preference

Stable Matching – Moving Towards *Algorithm Design*

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

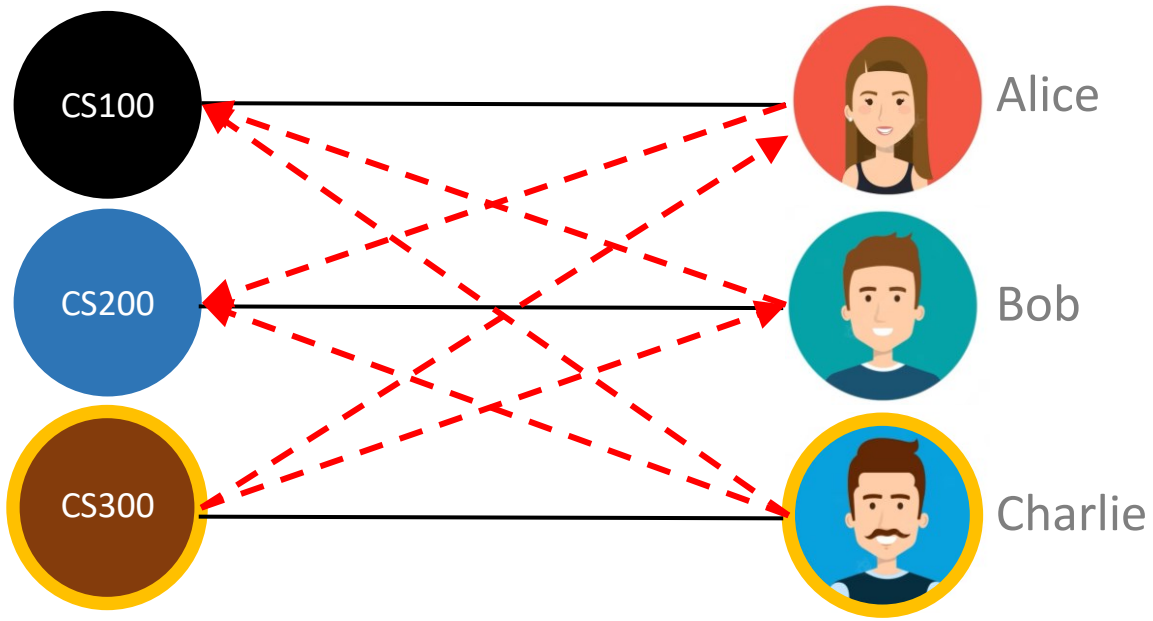
	1st	2nd	3rd
Alice	CS 200	CS 100	CS 300
Bob	CS 100	CS 200	CS 300
Charlie	CS 100	CS 200	CS 300

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



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

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

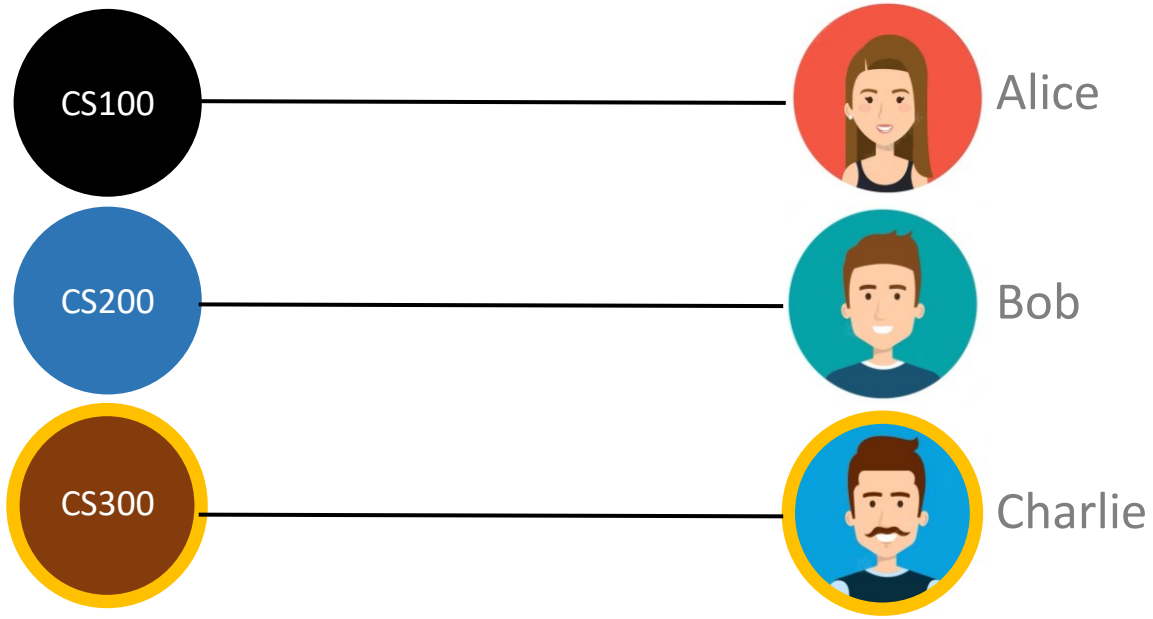
While Course *c* ∈ C is free
CS 300

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

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

What if ***a*** prefer ***c*** to ***c'*** ?

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



Let try new example – Worst Case Scenario

Stable Matching – Moving Towards *Algorithm Design*

1

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

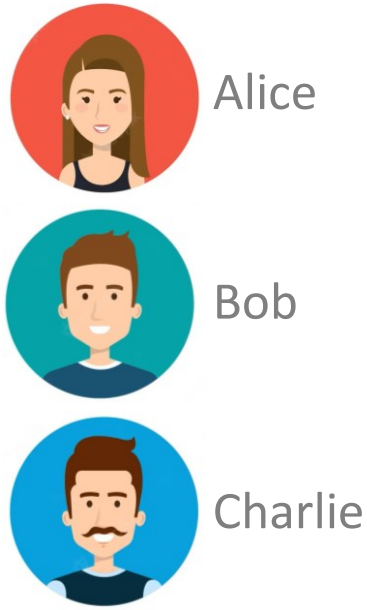
	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 100

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

If *a* is free then assign

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



Stable Matching – Moving Towards *Algorithm Design*

1

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

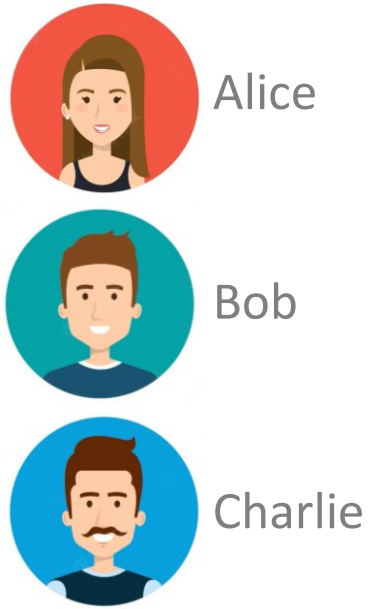
	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 100

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

If *a* is free then assign

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



Stable Matching – Moving Towards *Algorithm Design*

1

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

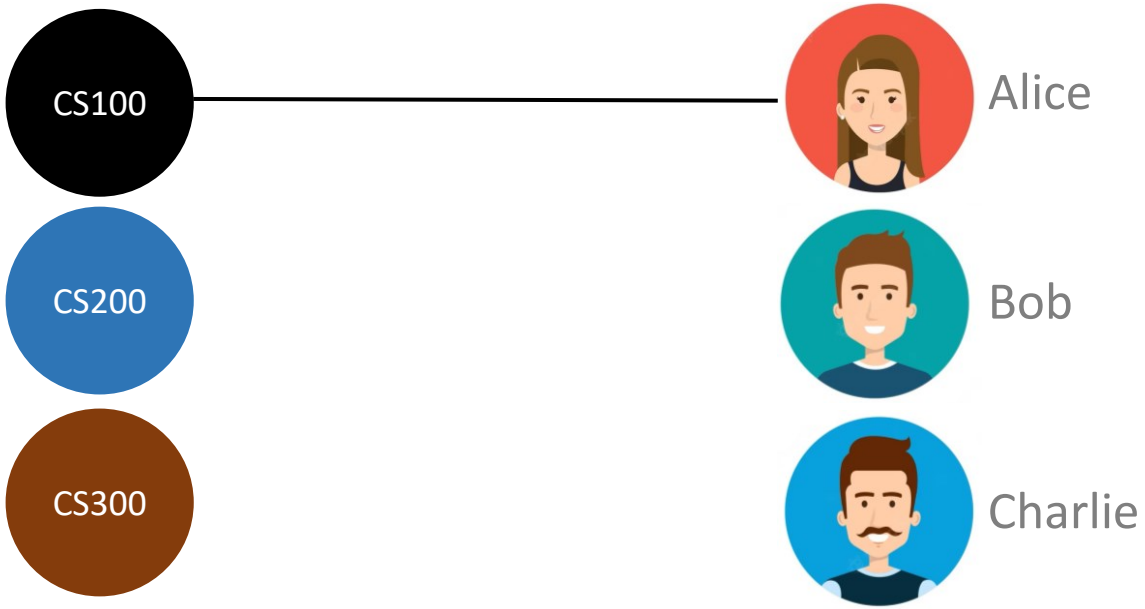
	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 100

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

If *a* is free then assign
(CS 100 – Alice)

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



Stable Matching – Moving Towards *Algorithm Design*

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

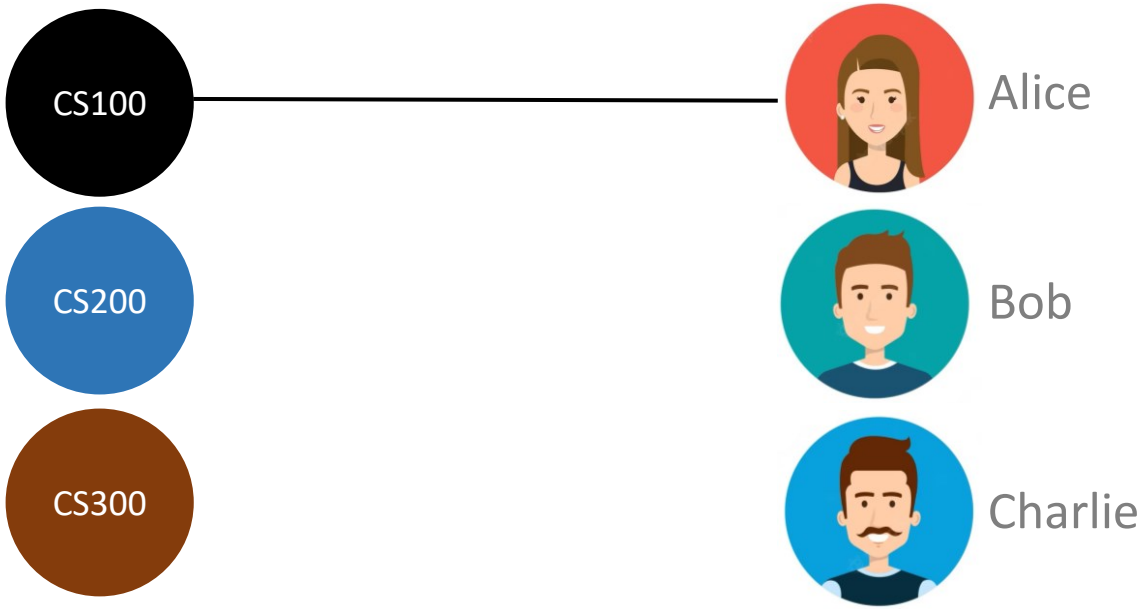
	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 200

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

If *a* is free then assign

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



Stable Matching – Moving Towards *Algorithm Design*

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

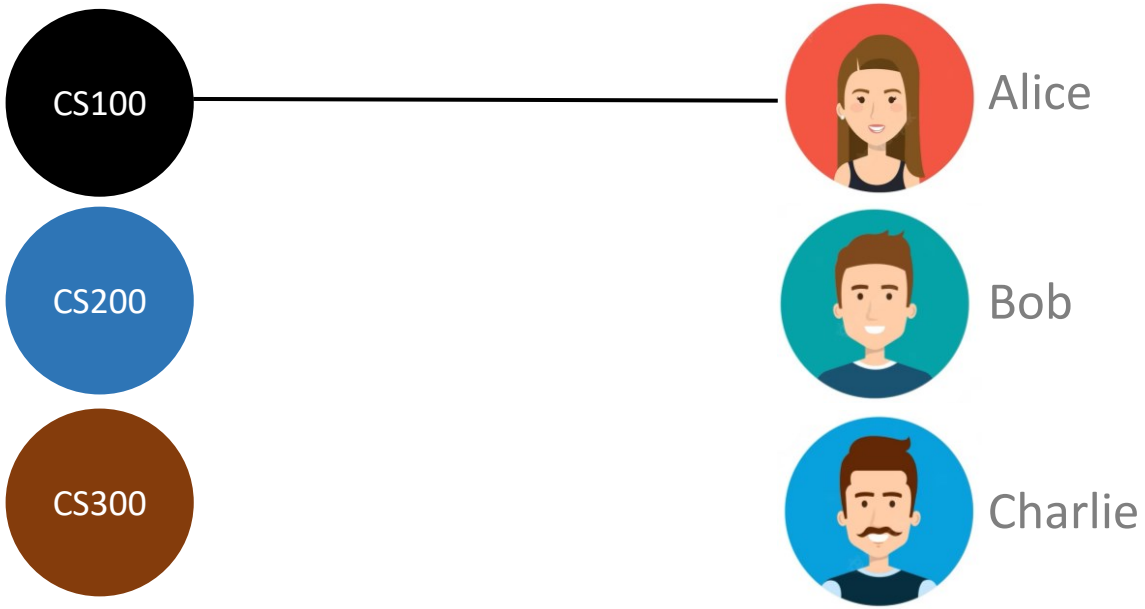
	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 200

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

If *a* is free then assign

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



Stable Matching – Moving Towards *Algorithm Design*

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

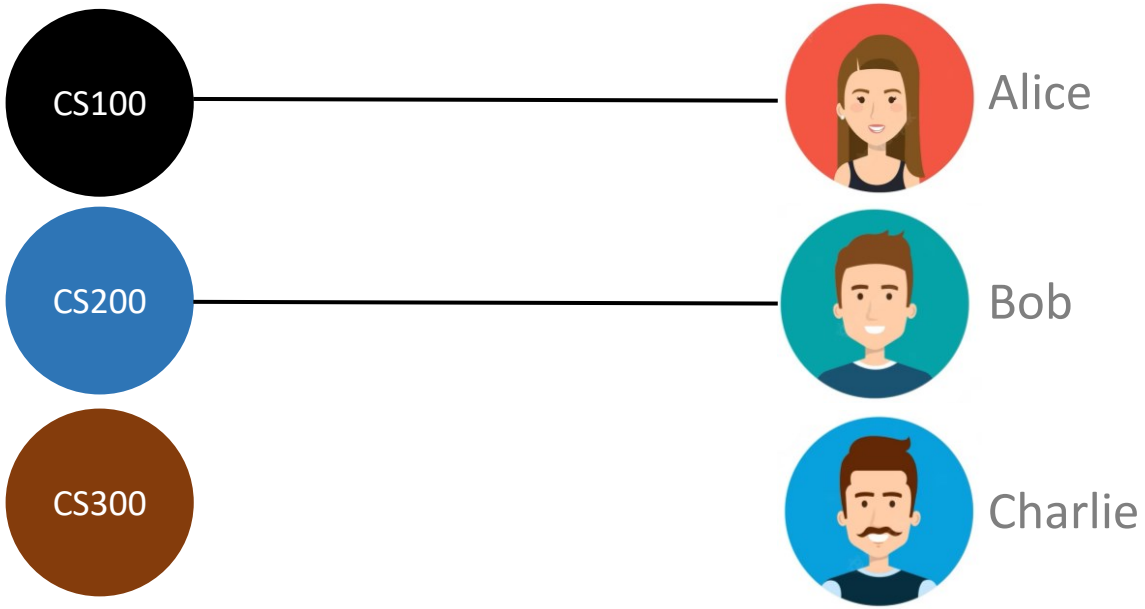
	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 200

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

If *a* is free then assign
(CS 200 – Bob)

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



Stable Matching – Moving Towards *Algorithm Design*

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

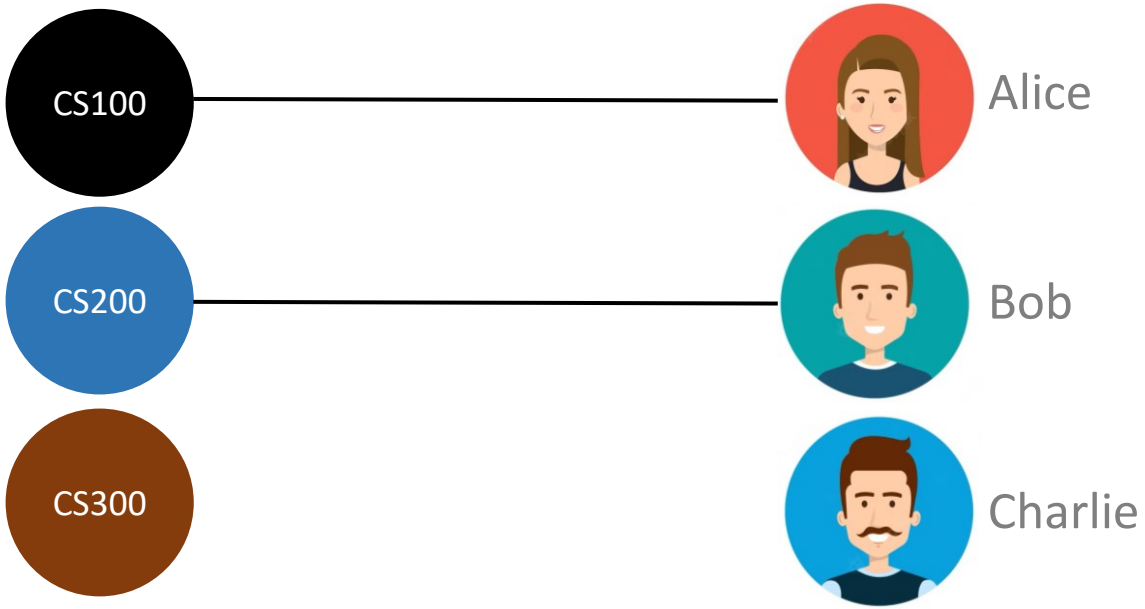
	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 300

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

If *a* is free then assign

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



Stable Matching – Moving Towards *Algorithm Design*

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

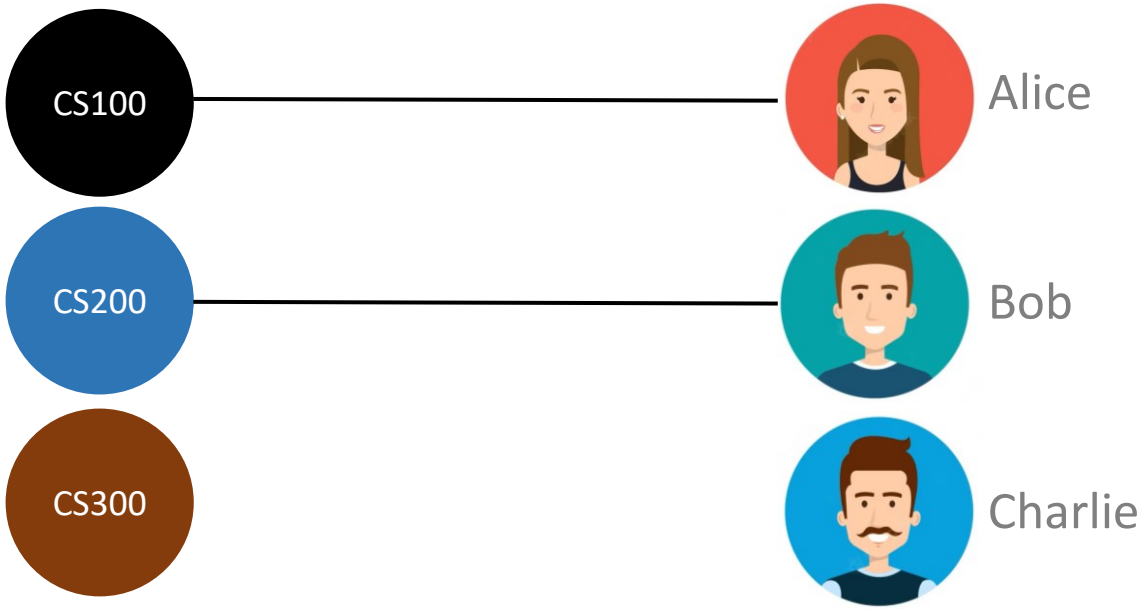
	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 300

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

If *a* is free then assign

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



Stable Matching – Moving Towards *Algorithm Design*

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

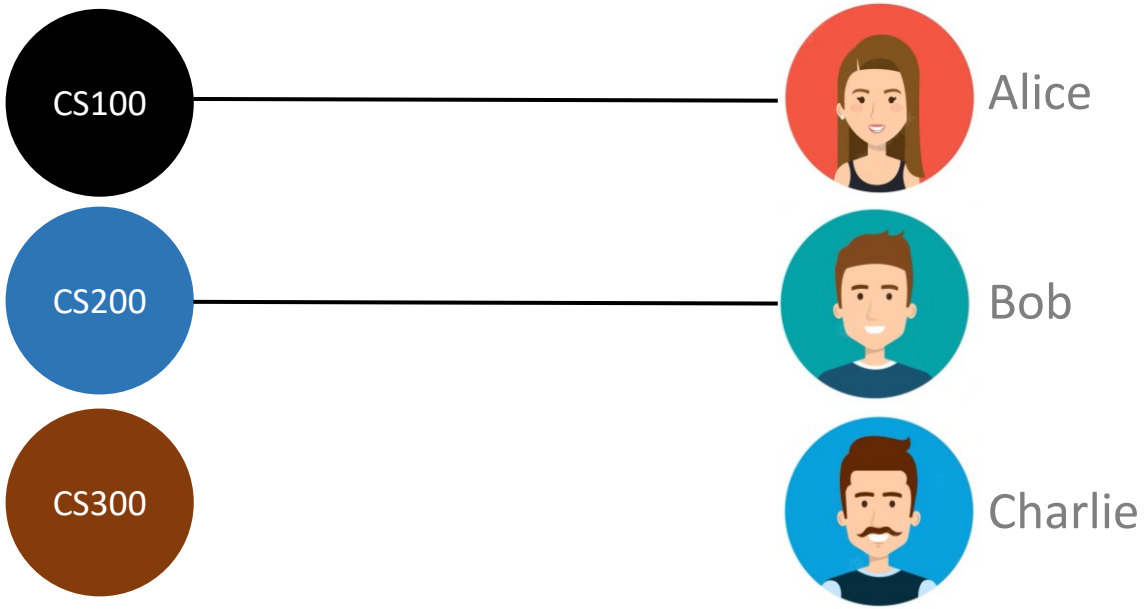
	1st	2nd	3rd
Alice	CS 200	CS 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

While Course *c* ∈ C is free
CS 300

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

If *a* is free then assign
Already assigned

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



Stable Matching – Moving Towards *Algorithm Design*

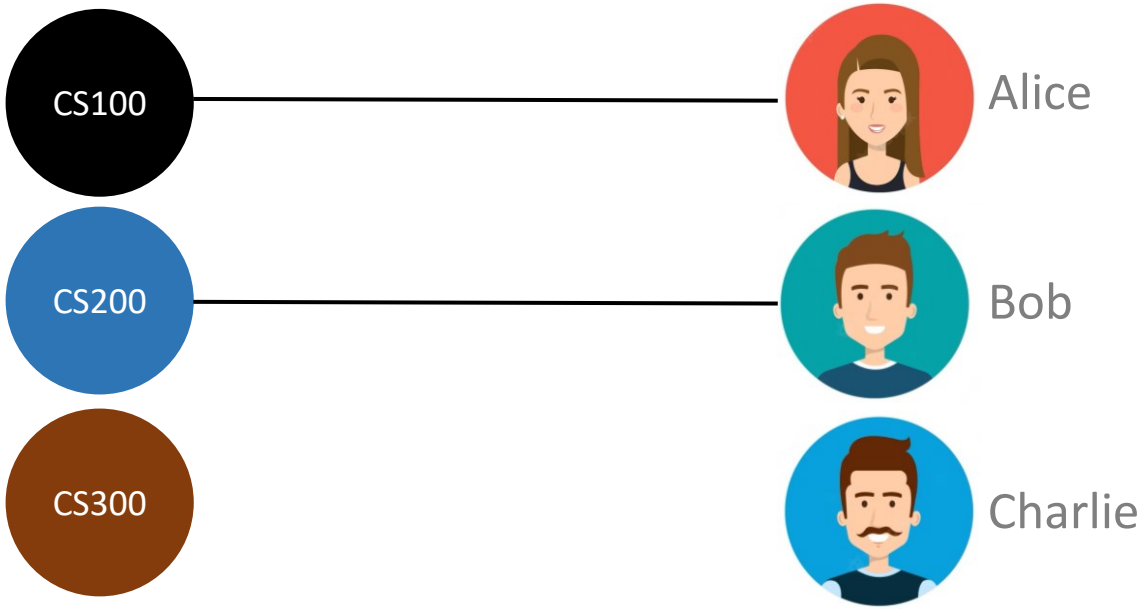
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

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

If *a* is free then assign
Already assigned



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*

	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

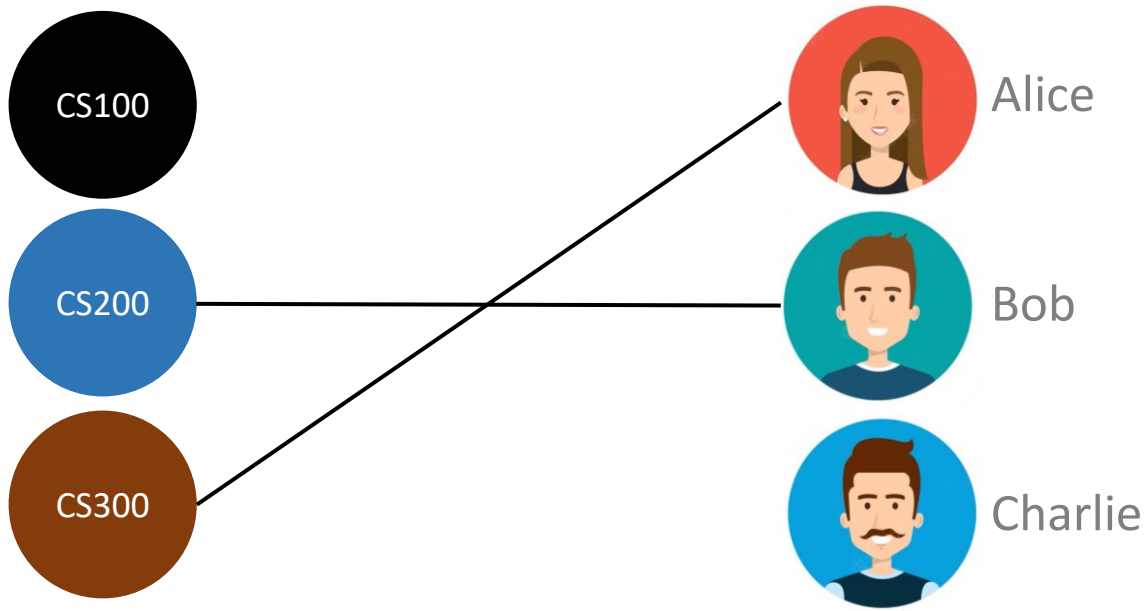
While Course *c* ∈ C is free
CS 300

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

If *a* is free then assign
Already assigned

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*

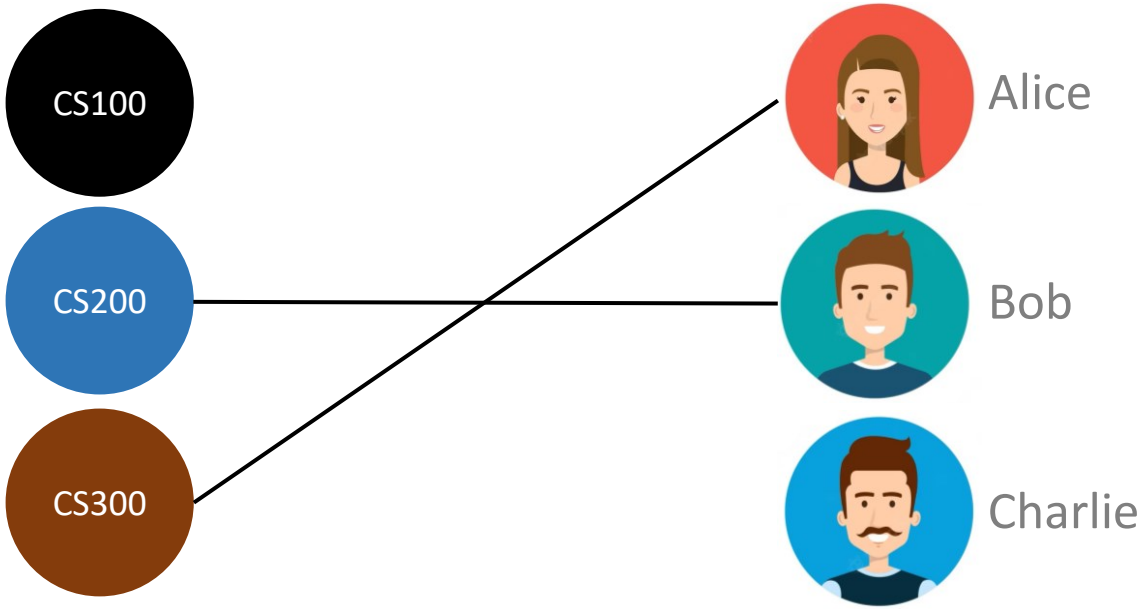
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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*

If *a* is free then assign



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*

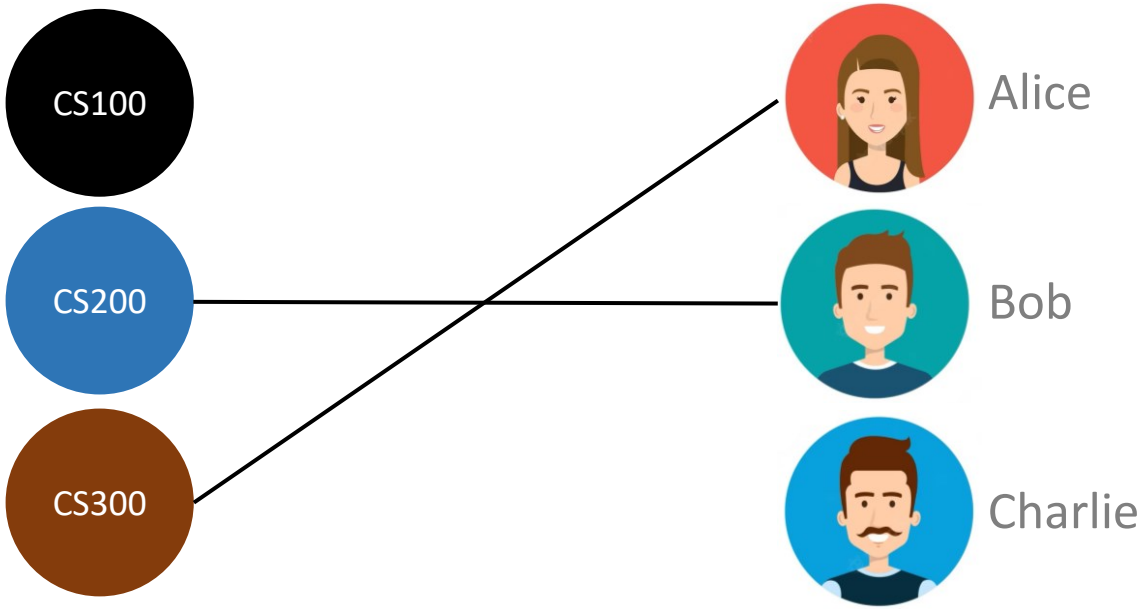
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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*
Bob

If *a* is free then assign



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*

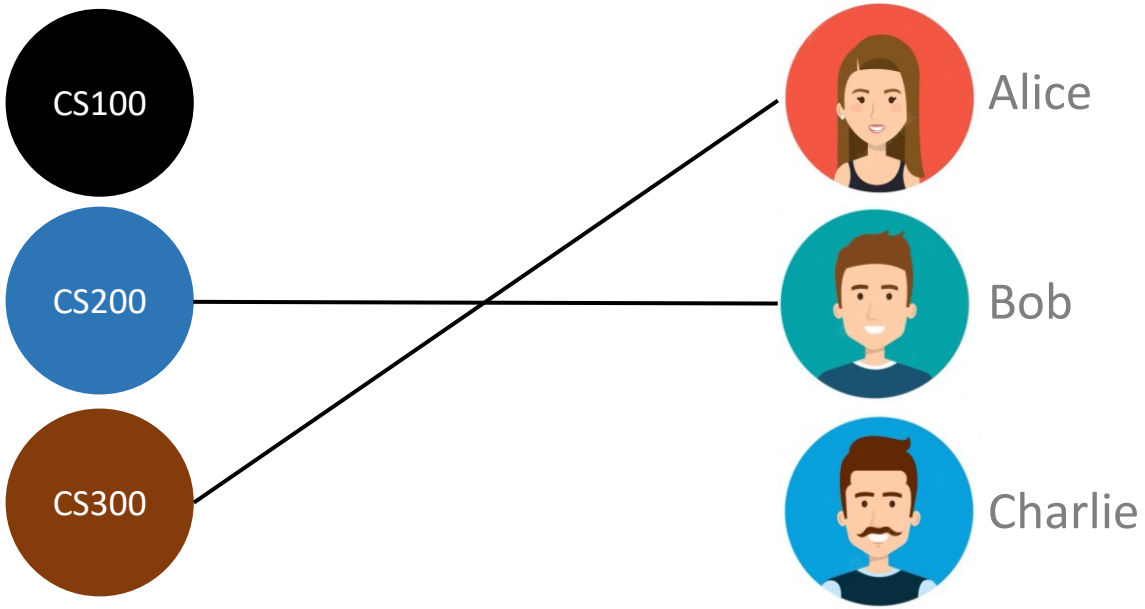
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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*
Bob

If *a* is free then assign
Already assigned



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*

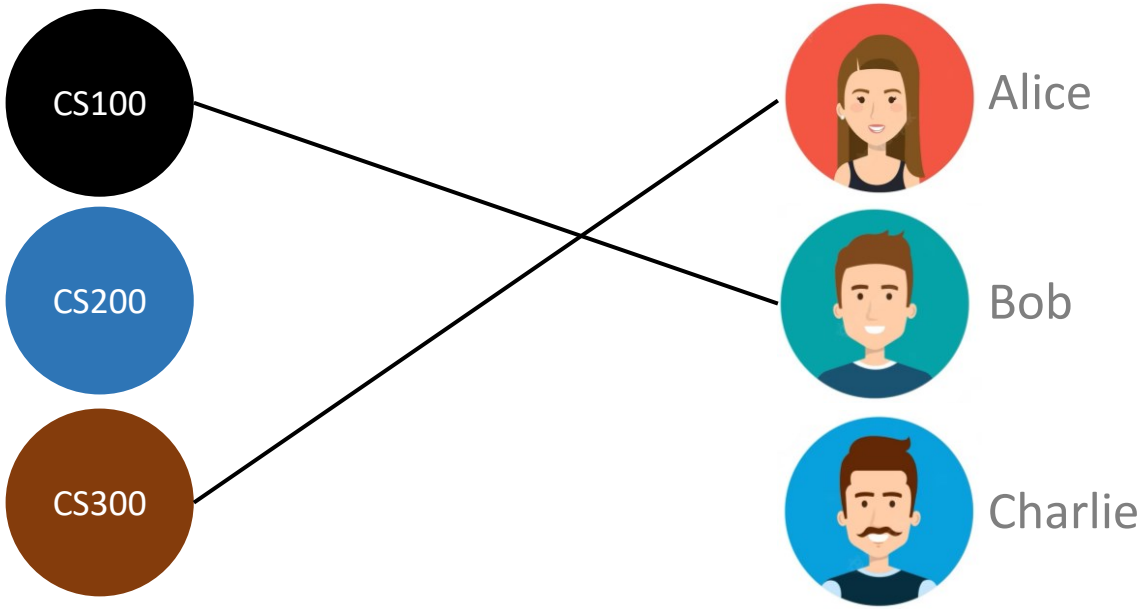
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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*
Bob

If *a* is free then assign
Already assigned



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*

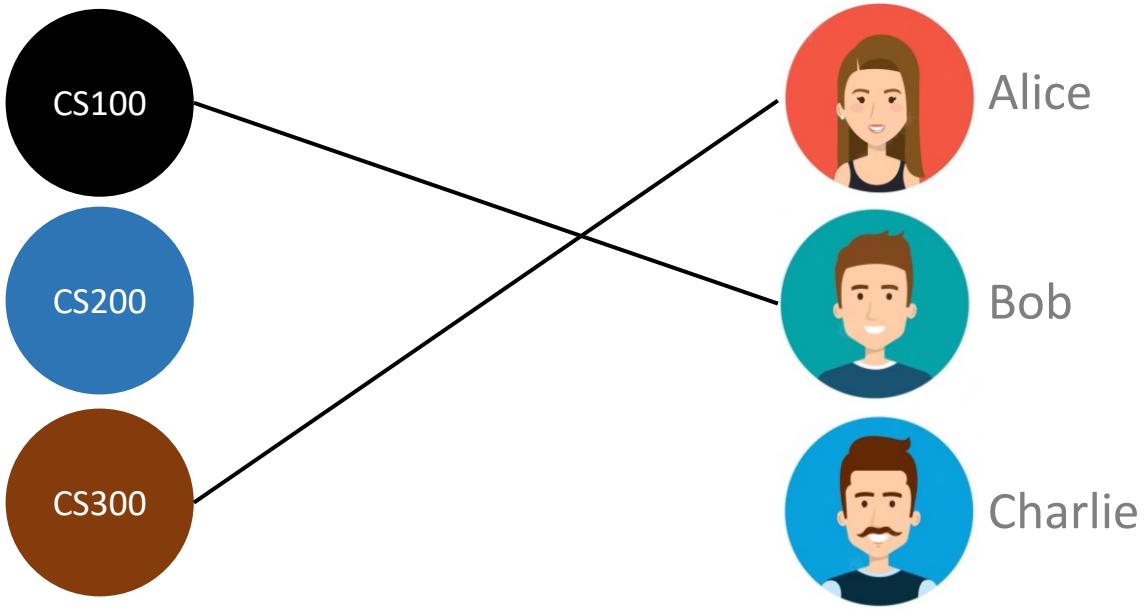
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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 200

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

If *a* is free then assign



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*

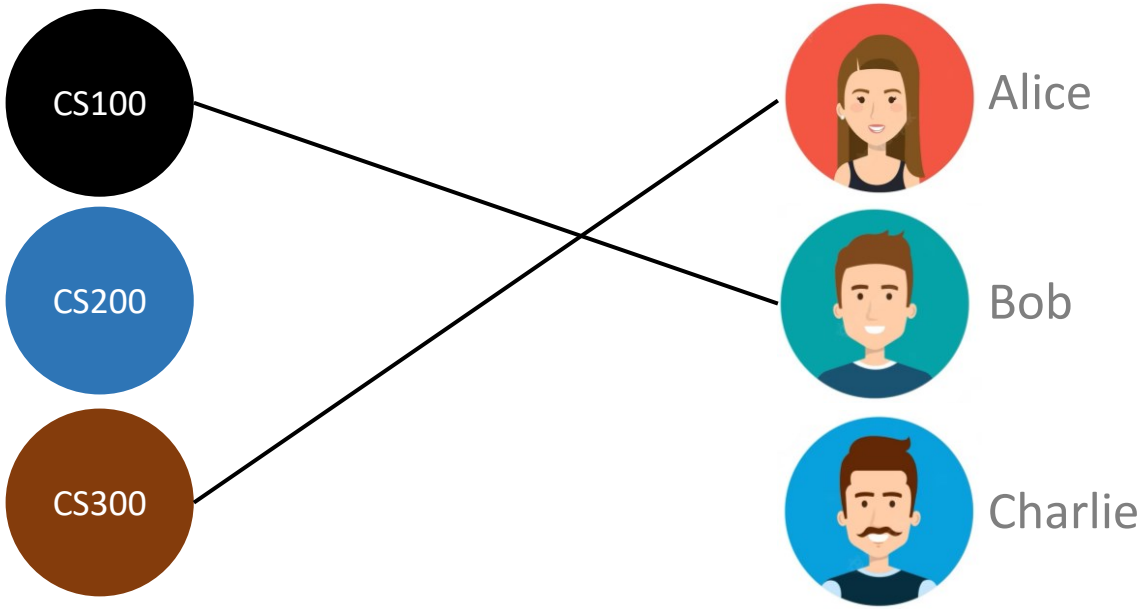
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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 200

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

If *a* is free then assign



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*

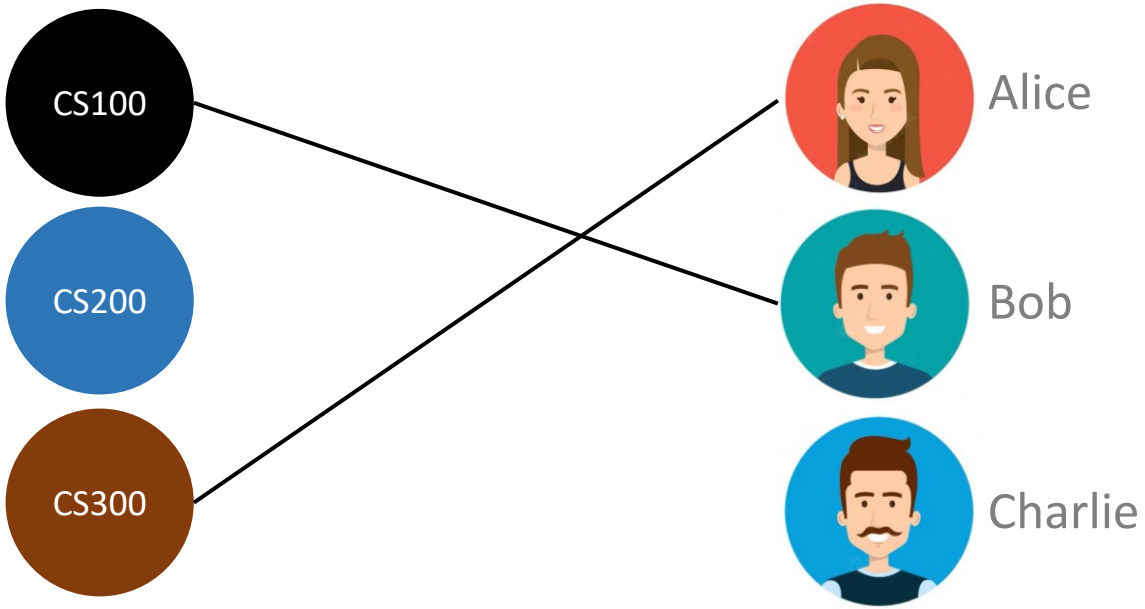
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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 200

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

If *a* is free then assign
Already Assigned



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*

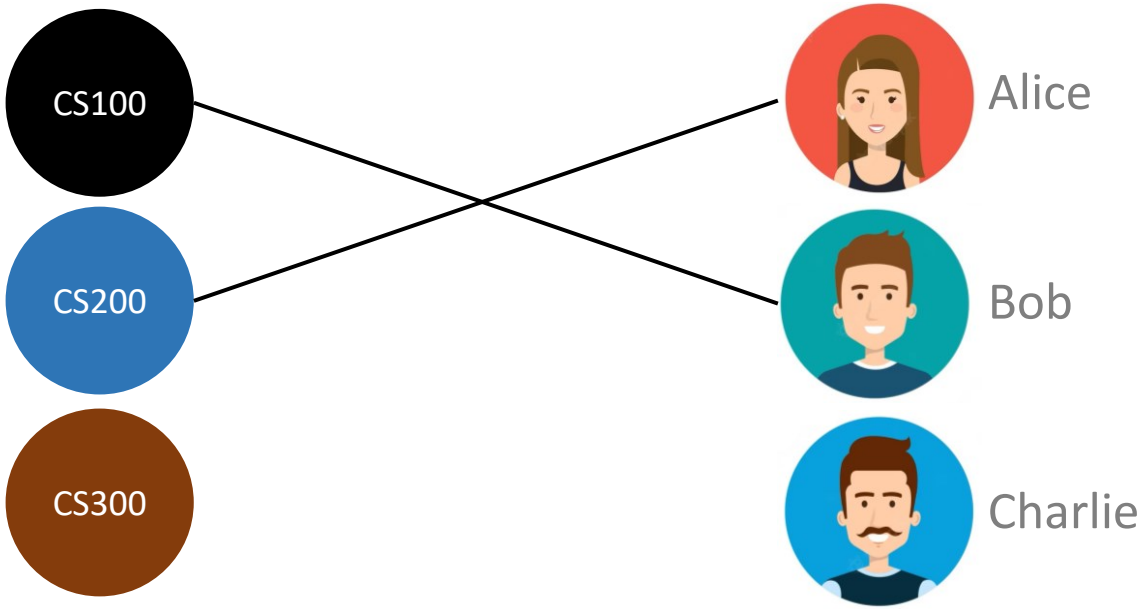
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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 200

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

If *a* is free then assign
Already Assigned



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*

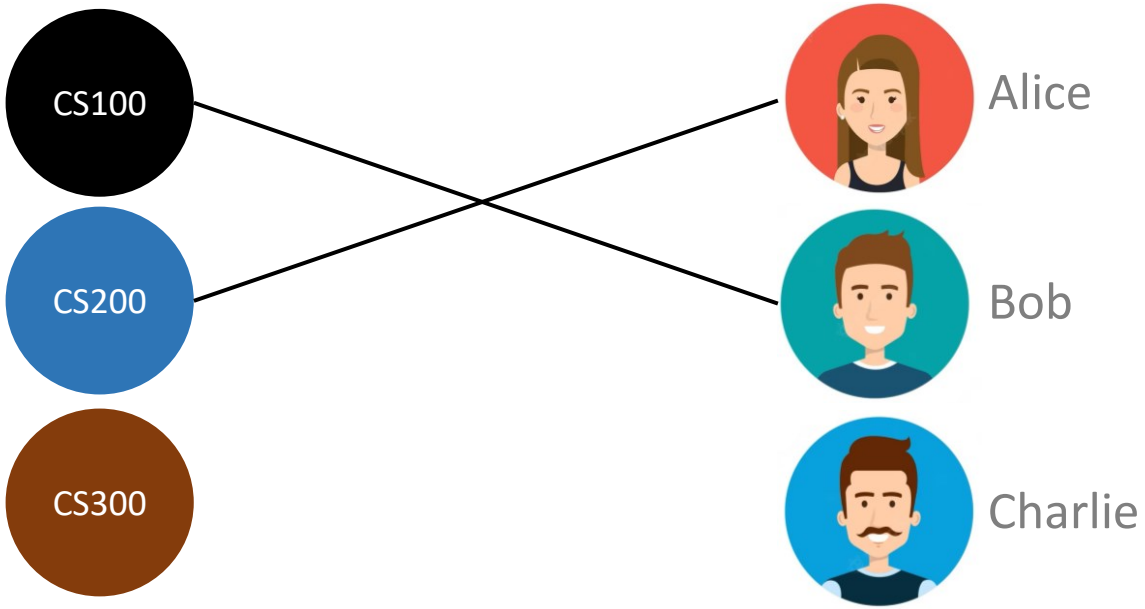
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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 300

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

If *a* is free then assign



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*

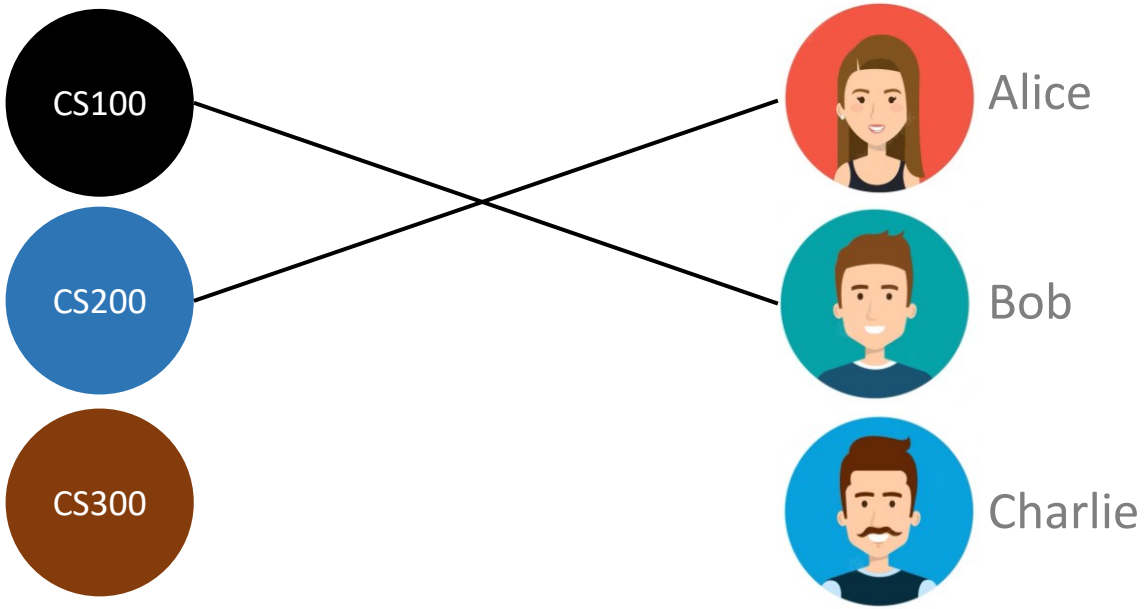
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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 300

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

If *a* is free then assign



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*

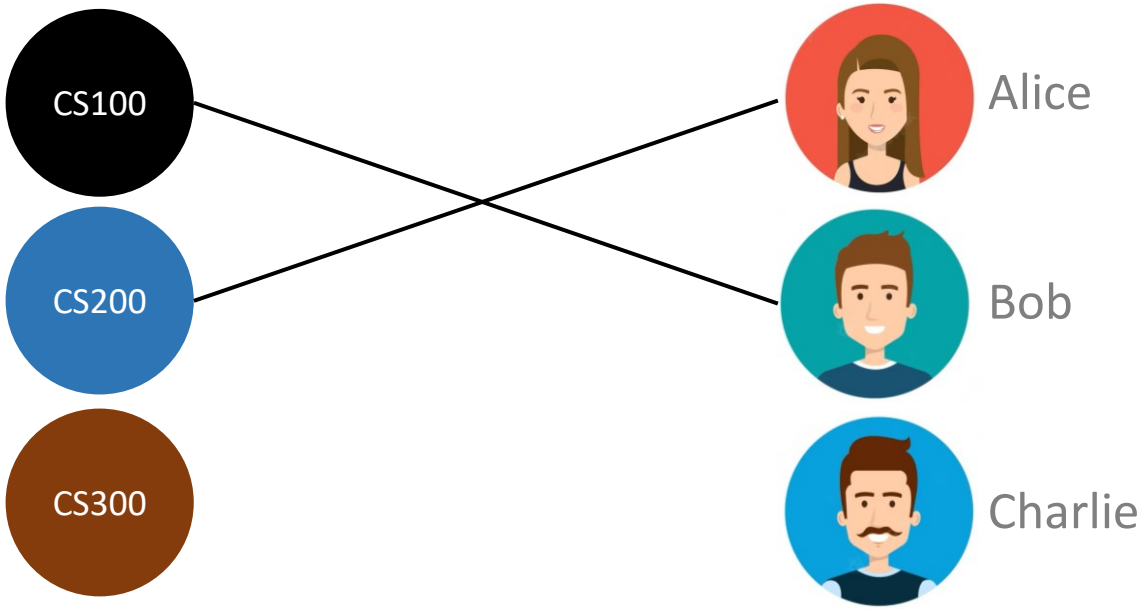
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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 300

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

If *a* is free then assign
Already Assigned



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*

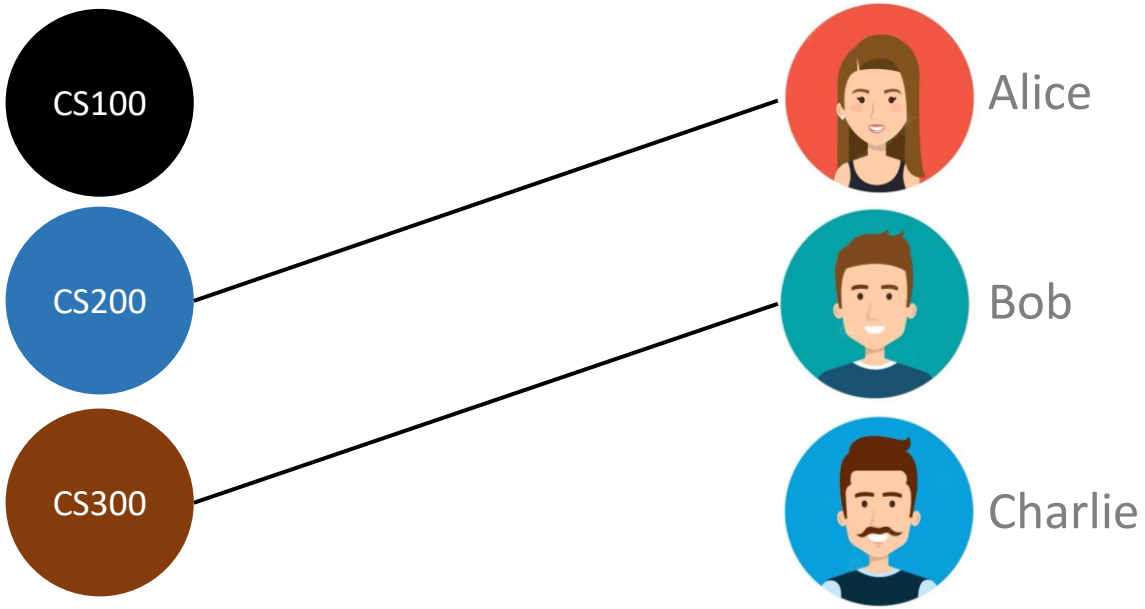
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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 300

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

If *a* is free then assign
Already Assigned



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*

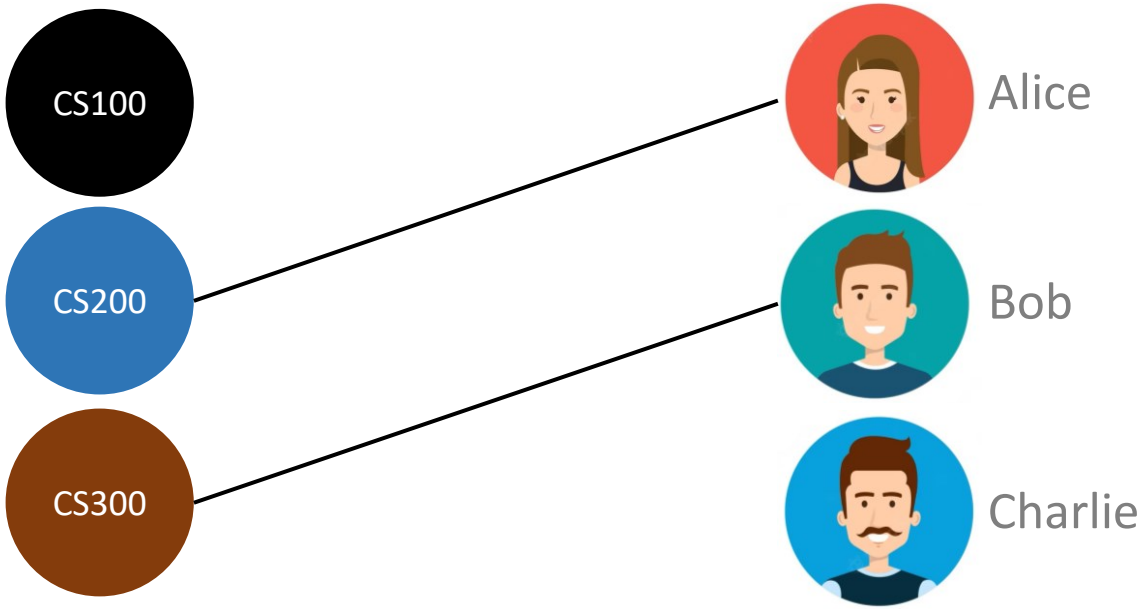
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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*

If *a* is free then assign



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*

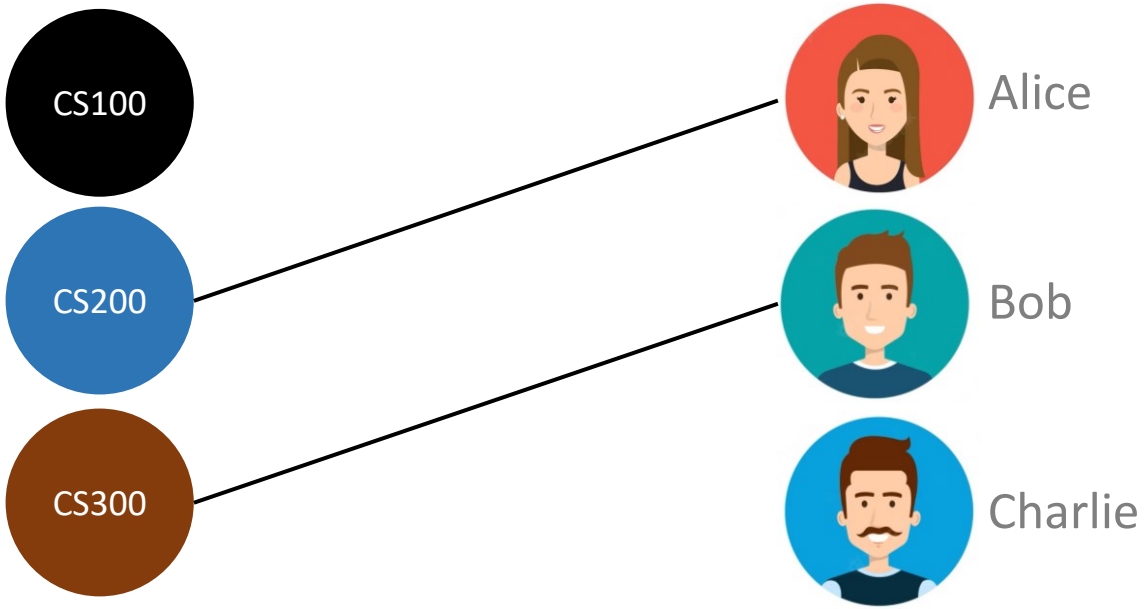
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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



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*

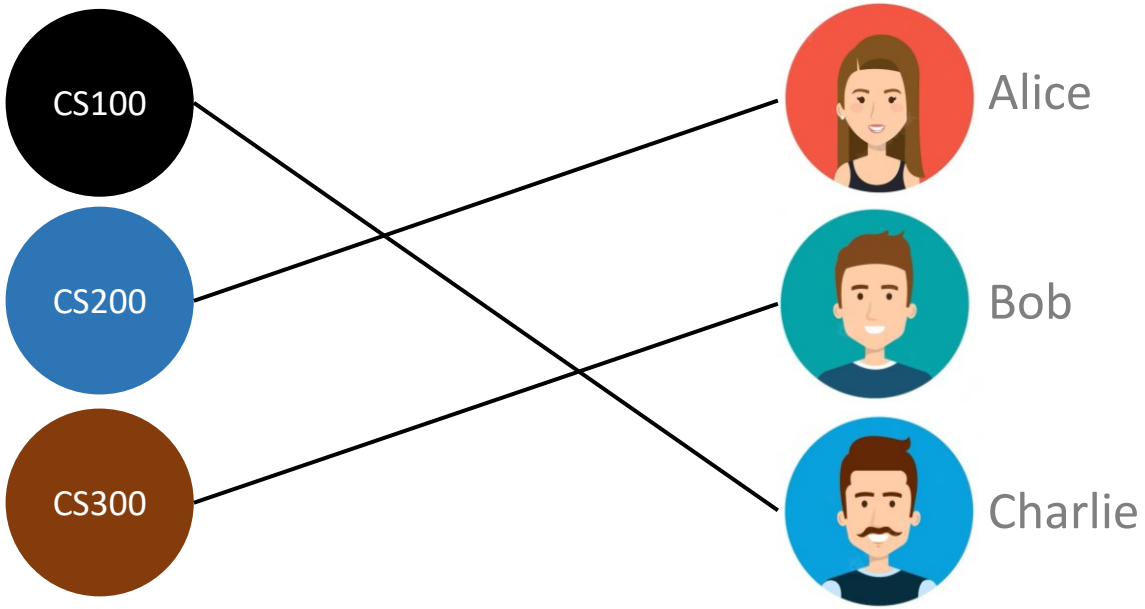
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	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

Stable Matching – Moving Towards *Algorithm Design*

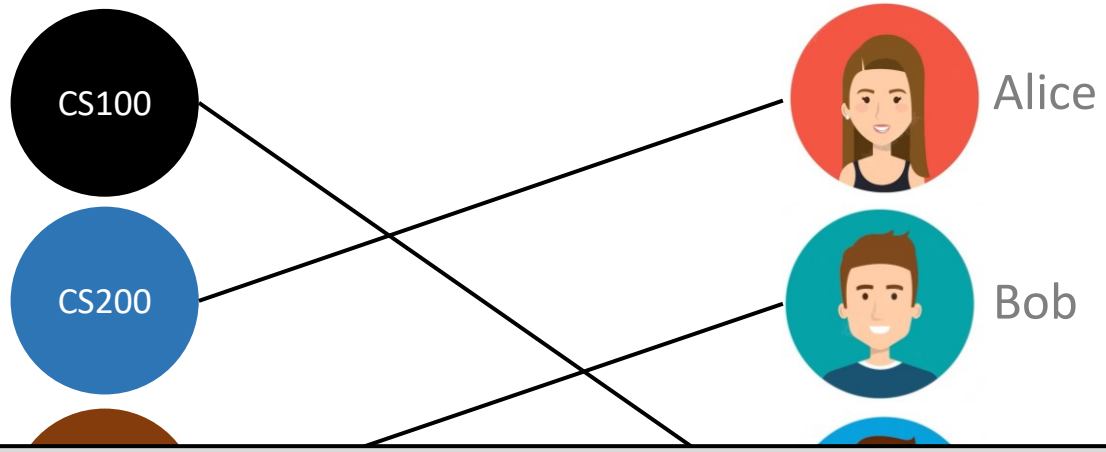
	1st	2nd	3rd
CS 100	Alice	Bob	Charlie
CS 200	Bob	Alice	Charlie
CS 300	Alice	Bob	Charlie

	1st	2nd	3rd
Alice	CS 200	CS 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



Observation 2. Once a TA applicant is assigned to a course (matched), the applicant never becomes unassigned (unmatched); only "trades up"

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

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 300	CS 100
Bob	CS 300	CS 100	CS 200
Charlie	CS 100	CS 200	CS 300

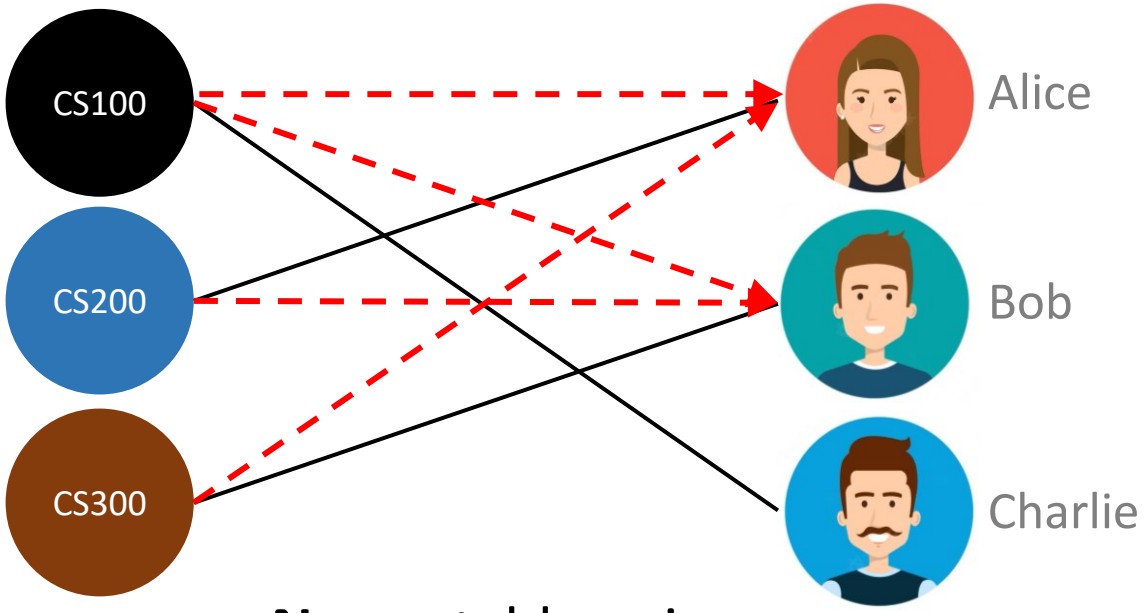
While Course *c* ∈ C is free
CS 100

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



No unstable pair

Gale-Shapely deferred acceptance algorithm

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

1 INITIALIZE M to empty matching
2 **While Course *c* ∈ C is free**
3 $a \leftarrow$ **Select Highest Preference *a* ∈ TA of *c to whom c has not yet offered***
4 **If *a* is free then assign**
5 Add $c - a$ to matching M
6 **Else if *a* prefers *c* to *c'* then assign *a* to *c* and *c'* gets free**
7 Replace $c' - a$ with $c - a$ in matching M
8 **Else if *a* prefers *c'* to *c* then c remains free**
9 a rejects offer made by c

11 RETURN stable matching M

Gale-Shapely deferred acceptance algorithm

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

1 INITIALIZE M to empty matching

2 WHILE (**Course *c* ∈ C is free and has not offered to every applicant**)

3 $a \leftarrow$ **Select Highest Preference *a* ∈ TA of *c to whom c has not yet offered***

4 IF (***a* has not received TAship offer before**)

5 Add $c - a$ to matching M

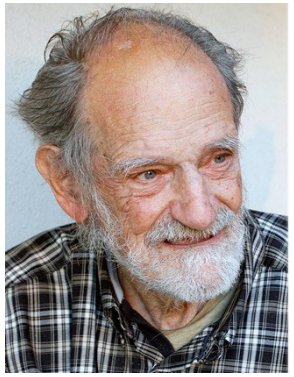
6 ELSE IF (***a* prefers *c* to a previously offered course by instructor c'**)

7 Replace $c' - a$ with $c - a$ in matching M

8 ELSE

9 a rejects offer made by c

11 RETURN stable matching M



Lloyd Shapley

In 2012, **Nobel Memorial Prize** in Economic Sciences was awarded to Lloyd S. Shapley "for the theory of stable allocations and the practice of market design".

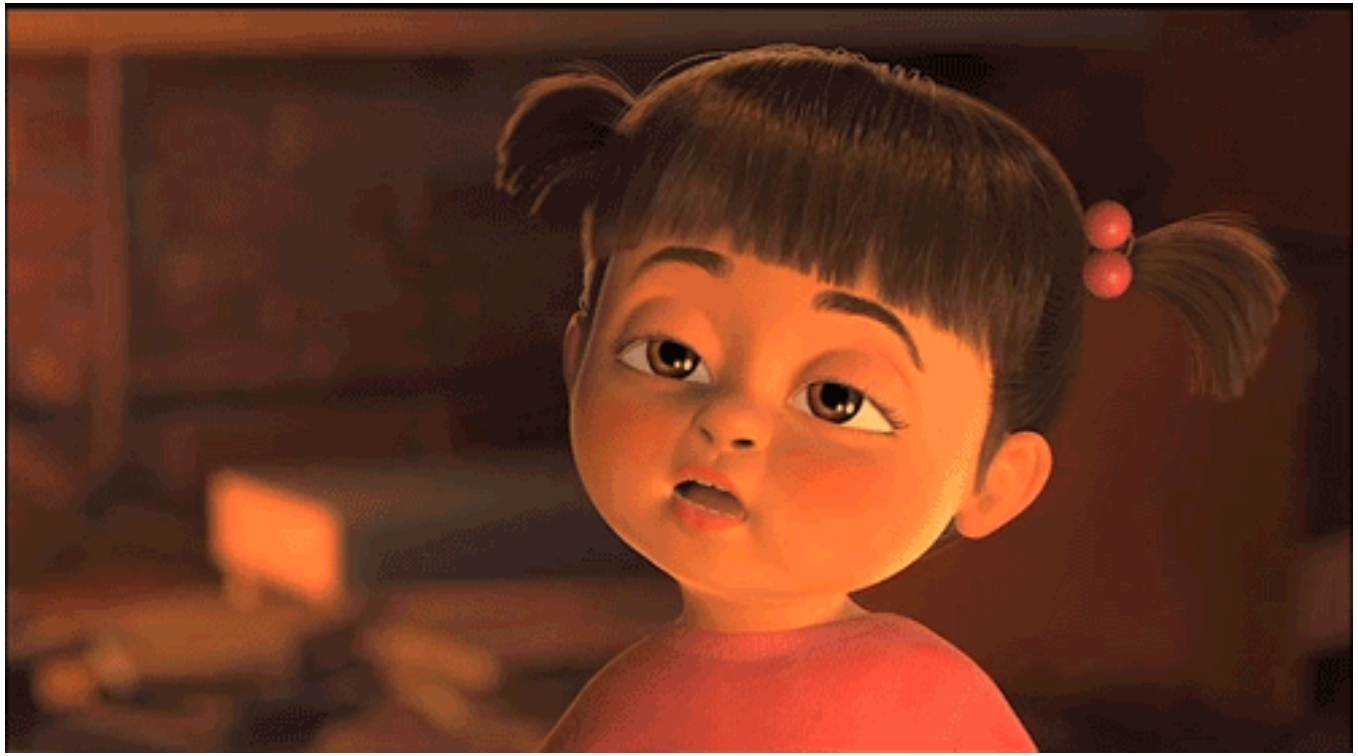
COLLEGE ADMISSIONS AND THE STABILITY OF MARRIAGE

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.

← original applications: college admissions and opposite-sex marriage

Thanks a lot



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