CSE 417 Autumn 2025

Lecture 1: Intro/Stable Matching

Nathan Brunelle and Glenn Sun

Intro/Logistics

About CSE 417

This is a course about **algorithms**.

Algorithm: A list of unambiguous instructions to solve a class of computational problems

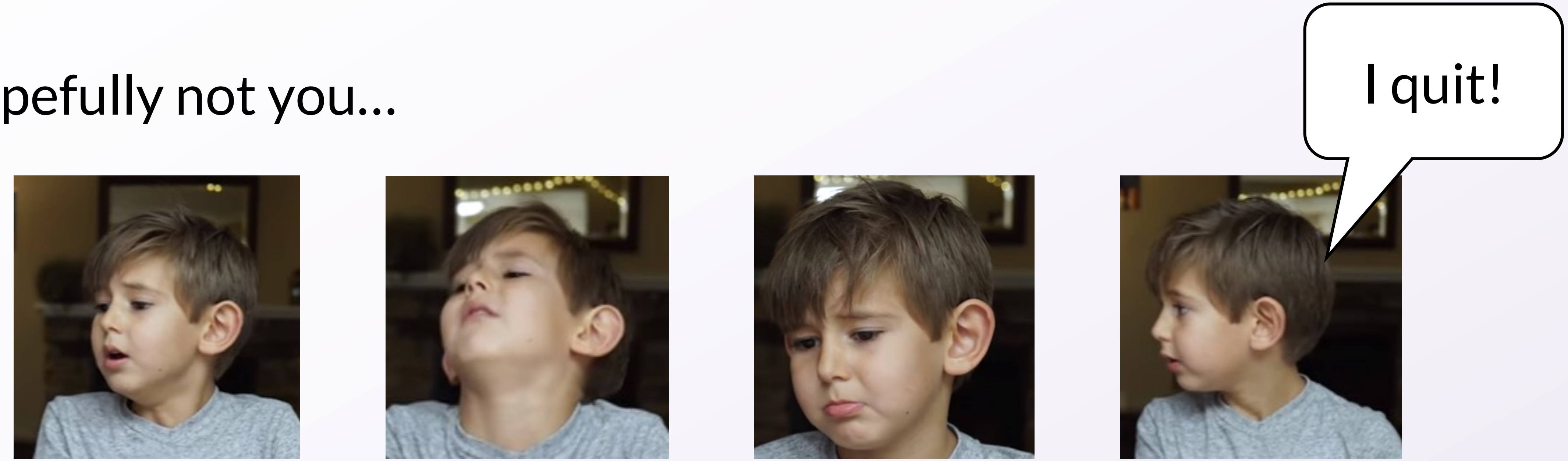
In this class, we will:

1. Learn some specific noteworthy algorithms
2. Practice techniques to design algorithms for new problems
3. Communicate clearly why our algorithms work

How it will sometimes feel

https://www.youtube.com/watch?v=cDA3_5982h8&t=245s

Hopefully not you…



Nathan Brunelle

Associate Teaching Professor

Faculty at UVA 2017-2023
At UW since autumn 2023

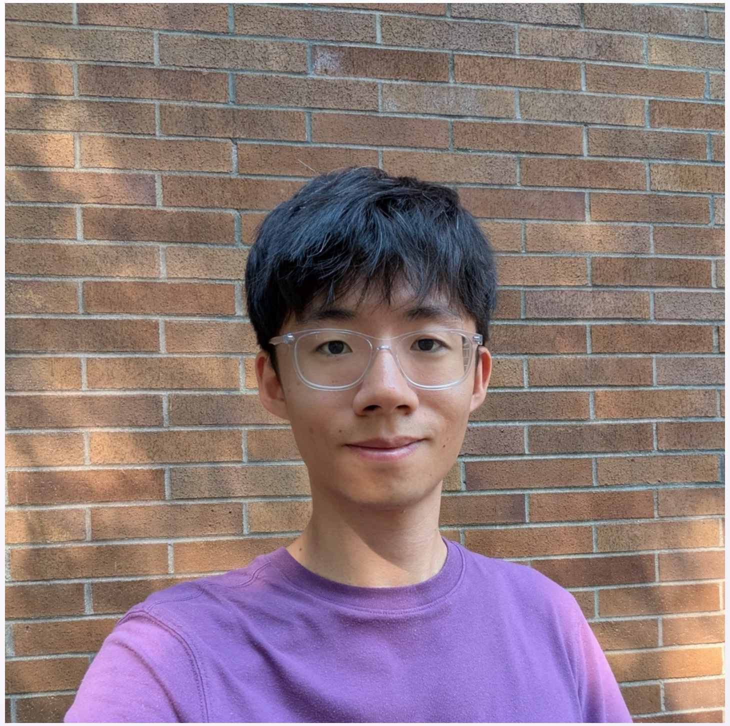
I've mostly taught courses on:
Discrete Math
Algorithms
Data Structures
Intro to Programming

Glenn Sun

3rd year PhD student

I study Boolean satisfiability (the topic of the last 2 weeks of this course!)

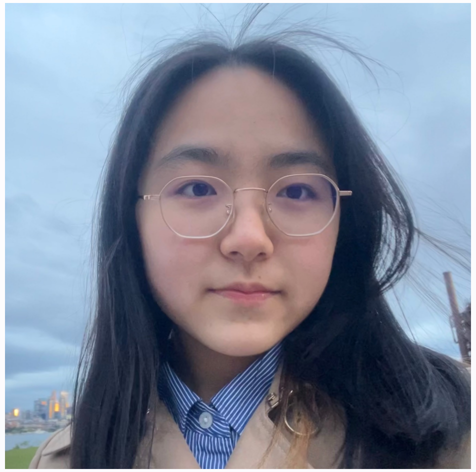
I've been teaching in various forms (math circles, HS summer camps, TAing) for ~6 years.



The TA Team

Evan

Katherine



Shayla

Zachary

Yuchen

Where to find things

- Canvas: Primary place for all course content, such as submitting homework, lecture recordings, etc.
- Ed Board: Questions and discussion with staff/other students

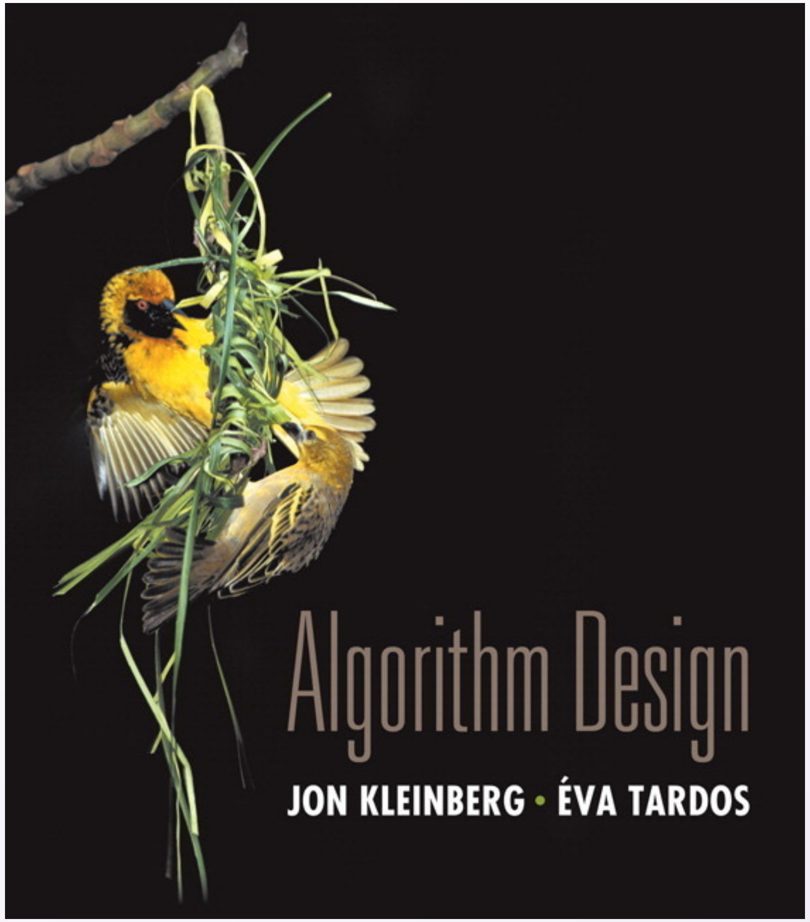
As necessary, items in Canvas will link to:

- cs.uw.edu/417: Public course content
- Gradescope: For a small number of assignments

Textbook

Not required.

Optional reference:
“**Algorithm Design**” by Jon Kleinberg and Éva Tardos



Homework

- **Pre-lecture reading + multiple choice concept check (10%):** before every lecture
- **Standard homework (35%):** 2 problems/week of algorithm design or similar longer-form mathematical work
- **Homework extensions (15%):** 1 problem/week of programming assignments, discussion posts, or theoretical extensions

HW 1 will release this Friday, due next Friday!

Resubmissions/Late policy

HW released!

↓ 7 days

Due date
(we start grading)

↓ 12 more days

Resubmission deadline
(solutions released)

Assessments

- **Quiz 1 (10%):** Friday, October 24, in class
- **Quiz 2 (10%):** Friday, November 21, in class
- **Final Exam (20%):** Monday, December 8

Please stress less!

- Slightly easier problems than homework
- Review session in the lecture slot before each assessment
- Standard reference sheets provided
- 1-page note sheet permitted

Grading scheme

Excellent (E)	Demonstrates near-mastery of skills
Satisfactory (S)	Got the main idea, but details wrong/missing or other notable room for improvement
Nearing Satisfactory (N)	Missed a key idea, but making substantial progress towards to a solution
Unsatisfactory (U)	Wrong direction, minimal progress, or failed to follow instructions

Note: N grades contribute positively to final grades in CSE 417.

Collaboration

Feel free collaborate with anyone in this class!

“Whiteboard-only” collaboration:

- Don’t take notes or take pictures when collaborating
- Write up solutions individually at home from memory

Cite your collaborators!

Do not ask past students or LLMs for homework help.

Office hours

Please come!

Nathan/Glenn OH:

- In-person OH: Lecturer holds after every class for 1 hr
- Online OH: Non-lecturer holds 30 min after every class for 1 hr

View TA's schedules on Canvas or cs.uw.edu/417/oh.

Read "**Guide to Office Hours**" on Canvas or cs.uw.edu/417/guides.

Anonymous feedback

feedback.cs.washington.edu

"nathan is so cool!"

"reading took me 1 hr :/"

"lecture today was really rushed :("

"more examples please!"

More details

For information about…

- ESNU to GPA minimum guarantees
- Accommodations and related procedures
- Details for everything we talked about so far
- Etc.

Please see the syllabus on Canvas or course website:

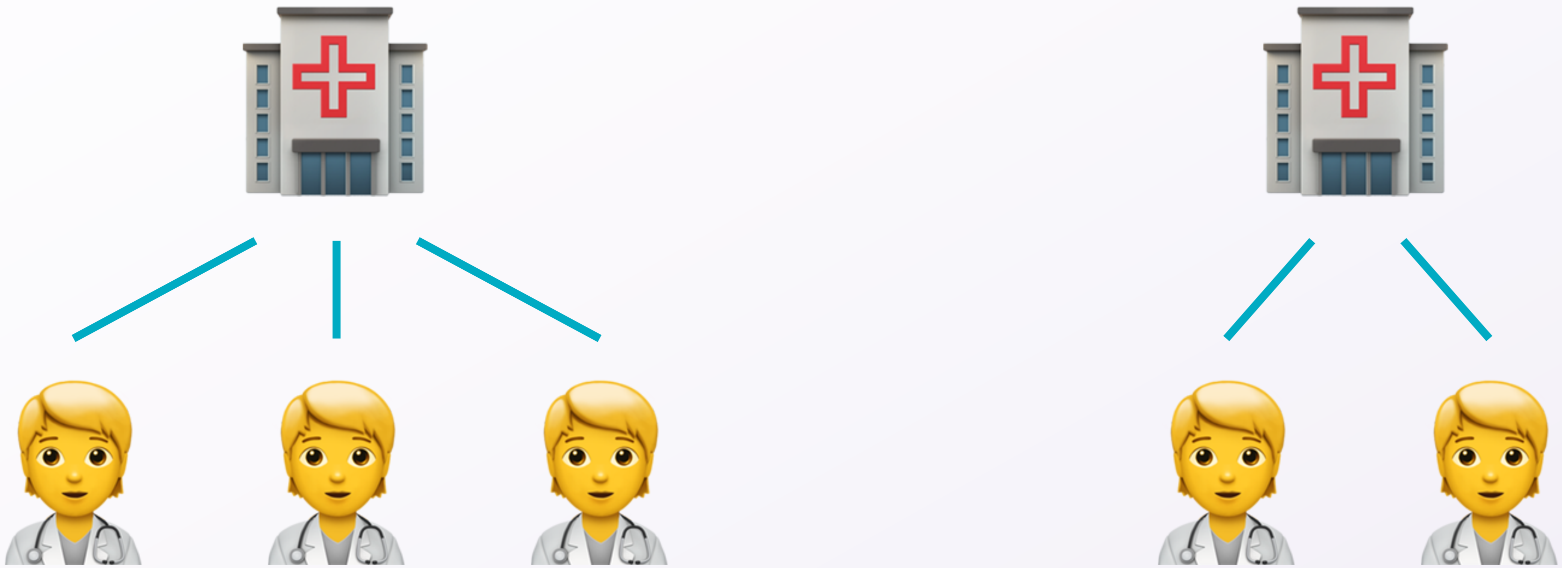
cs.uw.edu/417/syllabus

Stable Matching

Matching medical residents to hospitals (1/2)

Input: Preferences of hospitals and medical school residents

Goal: A self-reinforcing admissions process



Matching medical residents to hospitals (2/2)

Unstable pair: Resident r and hospital h are *unstable* if:

- r prefers h to their assigned hospital, and
- h prefers r to one of its admitted residents.

Stable assignment: Assignment with no unstable pairs

- Individual self-interest will prevent any applicant/hospital side deal from being made

(Possible, but a bit more complicated—will discuss in HW2.)

Simplification: Stable matching problem

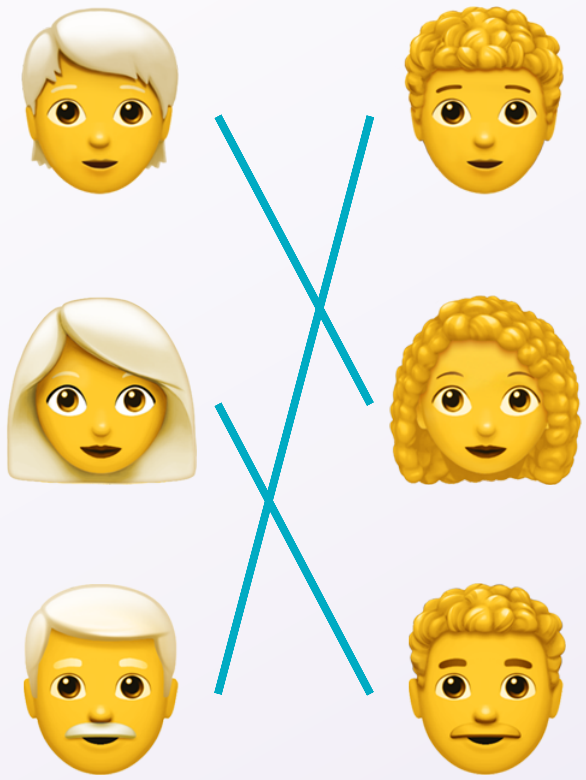
Input: Preferences of two groups of same size

Goal: Find a ***stable matching*** (perfect matching with no unstable pairs)

Perfect matching: One-to-one assignment between two groups of the same size

Unstable pair: A pair (x, y) is *unstable* if:

- x prefers y to its assigned partner, and
- y prefers x to its assigned partner.



Stable matching examples (1/3)

Example input:

1:	A	> B	>	C
2:	B	> A	>	C
3:	A	> B	>	C

A:	2	> 1	>	3
B:	1	> 2	>	3
C:	1	> 2	>	3

Q: Is matching (1, C), (2, B), (3, A) stable?

A: No, (1, B) is an unstable pair.

Stable matching examples (2/3)

Example input:

1:	A	>	B	>	C
2:	B	>	A	>	C
3:	A	>	B	>	C

A:	2	>	1	>	3
B:	1	>	2	>	3
C:	1	>	2	>	3

Q: Is matching (1, A), (2, B), (3, C) stable?

A: Yes (can brute-force check, or see that the only number who could improve is 3, but all letters dislike 3 the most.)

Stable matching examples (3/3)

Example input:

1:	A	>	B	>	C
2:	B	>	A	>	C
3:	A	>	B	>	C

A:	2	>	1	>	3
B:	1	>	2	>	3
C:	1	>	2	>	3

Q: Is matching (1, B), (2, A), (3, C) stable?

A: Also yes!

Gale–Shapley algorithm

Group P proposes, group R receives proposals.

1. **while** there is a free proposer $p \in P$ **do**
2. Let r be the top person on p's preference list that p has not yet proposed to.
3. **if** r is also free **then**
4. Have r accept p.
5. **if** r is paired to p' but prefers $p > p'$ **then**
6. Have r accept p and also leave p'.
7. **return** all matches

Example:

1: A > B > C

2: B > A > C

3: A > B > C

A: 2 > 1 > 3

B: 1 > 2 > 3

C: 1 > 2 > 3

Tentative pairs:

1	A
2	B
3	C

Questions about Gale–Shapley (1/3)

Q: How do we know that the output is actually a stable matching?

Q: How do we know that the while loop always terminates?

Q: Line 2 said to take the top receiver r that p has not yet proposed to — what if p proposed to every r already? Is that impossible?

A: Wait for Lecture 4!

Questions about Gale–Shapley (2/3)

Q: Is the Gale–Shapley algorithm a fair algorithm?

A: Potentially not. Recall our examples.

Stable matching 1:
(**found** by G–S)

1:	**A**	>	B	>	C	A:	2	>	**1**	>	3
2:	**B**	>	A	>	C	B:	1	>	**2**	>	3
3:	A	>	B	>	**C**	C:	1	>	2	>	**3**

Stable matching 2:
(**not found** by G–S)

1:	A	>	**B**	>	C	A:	**2**	>	1	>	3
2:	B	>	**A**	>	C	B:	**1**	>	2	>	3
3:	A	>	B	>	**C**	C:	1	>	2	>	**3**

Questions about Gale–Shapley (3/3)

Proposer optimality theorem: The Gale–Shapley algorithm always finds the unique stable matching that is both **best for proposers** and **worst for receivers**.

(Some sources will state G–S for men/women and marriage, but this algorithm **should not** be used for this application.)

Your To-Do List

- Say hello on Ed!
- Read the syllabus fully, on Canvas and cs.uw.edu/417
- Complete the pre-class reading and concept check on Canvas before Friday's lecture

Homework 1 will be released after Friday's lecture!