CS 7800: Advanced Algorithms

Instructor: Soheil Behnezhad (call me Soheil)

Research:

- Graph algorithms
- Sublinear algorithms / algorithms for big data

Office: WVH 348

Office Hours: Tue 3:30pm-4:30pm in my office.

Teaching Assistant: Amir Azarmehr

Office Hours: TBD

Algorithms

- What is an algorithm?

An explicit, precise, unambiguous, mechanically-executable sequence of elementary instructions for solving a computational problem.

-Jeff Erickson

- Essentially all computer programs (and more) are algorithms for some computational problem.

Algorithms

- What is algorithms?

    The study of how to solve computational problems.

    - Abstract and formalize computational problems
    - Identify useful algorithmic tools for solving computational problems
    - Analyze and compare algorithms
        - This class: correctness, running time, space usage
        - Beyond: parallelism, robustness, simplicity, extensibility

Why study algorithms?

- **Improve problem solving:**
  - How/why do algorithms really work?
  - How to attack new problems?
  - Which design techniques work well?
  - How to compare different solutions?
  - How to know if a solution is the best possible?

- **Improve communication:**
  - How to explain solutions?
  - How to convince someone that a solution is correct?
  - How to convince someone that a solution is best?

Why study algorithms?

- **Improve the world:**
  - Algorithms are pervasive
  - Can increase productivity
  - Can increase social utility
  - Can increase fairness

Why study algorithms?

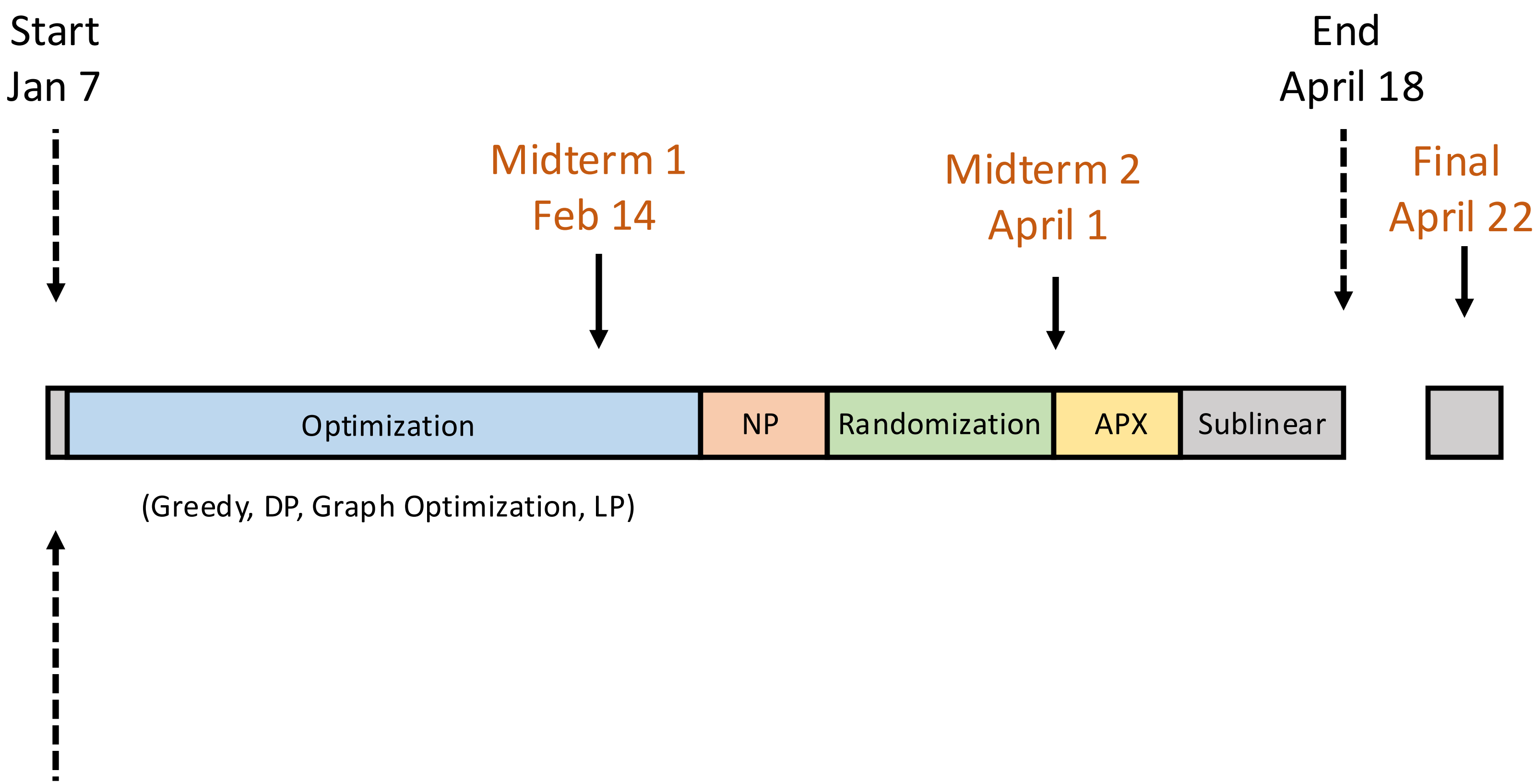
- **Inventors we all admire**

Edison/Tesla	electricity
Guttenberg	printing press
Edward Jenner	smallpox vaccine

- **Many modern inventions are algorithmic**

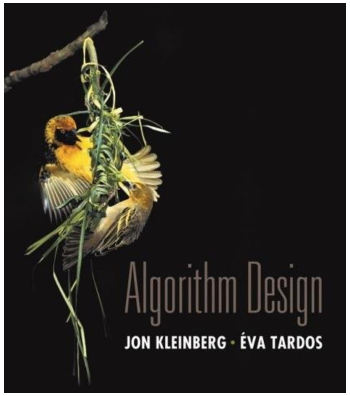
Dijkstra	Shortest path	⇒ internet routing
Cooley, Tukey	Fast Fourier Transform	⇒ audio/image processing
Rivest-Shamir-Adleman	RSA protocol	⇒ securing internet
Knuth	Text search	⇒ word processors
Hamming/Shannon	Error-correcting code	⇒ CDs, communications
Page	PageRank	⇒ Google search

Course Structure



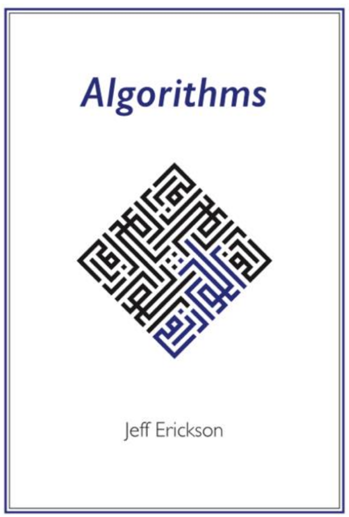
Prereq:
Asymptotic notation, Graphs, Proof by induction, Mathematical reasoning

Resources



Algorithm Design by Kleinberg and Tardos

We follow this book closely for the first half.



Algorithms by Jeff Erickson
(freely available)

Useful for review, alternative perspective, and some advanced material.

Grading

- 30% Homework Assignments
- 40% Midterms (20% each)
- 30% Final Exam
- +5% Active Participation

  (engaging and asking questions in class)

Homework

- 30% of grade
- There will be 5 HWs.
- Lowest HW score will be dropped from your grade
  - The remaining 4 HWs are weighted equally
- Homework will have a mix of mathematical and algorithmic questions but no programming

- **Start homework early!!!**

Homework Policies

- **Collaboration is strongly encouraged**
  - You can collaborate with up to two other students per HW
  - You should list your collaborators on your solution set
  - You must write all solutions by yourself (no written material should be shared)

- **Using Internet/LLMs for solving HW problems is strongly prohibited**
  - I take this very seriously!

- **We reserve the right to ask you to explain any submitted material in person**

Homework Policies

- Solutions must be typeset in LaTeX!
  - Many resources and editors available (Overleaf, TexShop, TexStudio)
  - We will provide problem set source .tex file + PDF

The Not So Short
Introduction to $\LaTeX\ 2_\varepsilon$

Or $\LaTeX\ 2_\varepsilon$ in 157 minutes

by Tobias Oetiker

Hubert Partl, Irene Hyna and Elisabeth Schlegl

Version 5.06, June 20, 2016

Homework Policies

- You should submit your HW on Gradescope!
  - Integrated with Canvas
  - Tag your pages! Penalty will be assessed for failing to do so

gradescope

Academic Integrity Policies

- You are encouraged to work with your classmates on the homework problems.
  - You may not use the internet
  - You may not contact students/people outside of the class
- **Homework Collaboration Policy:**
  - You must write all solutions by yourself
  - You may not share any written solutions
  - You must state all your collaborators
  - We reserve the right to ask you to explain any solution
- **Maintain highest academic integrity standard throughout, including all tests and assignments**
  - Northeastern Academic Integrity Policy

Course Webpage

http://behnezhad.com/cs7800-spr25/

Advanced Algorithms (Spring 2025)

Course	CS 7800 Advanced Algorithms
Semester	Spring 2025
Instructor	Soheil Behnezhad (Office hours: Tuesdays 3:30pm-4:30pm at WVH 348.)
Meeting Time	Tuesdays and Fridays 9:50am - 11:30am in Snell Library 049
Office Hours	Tuesdays 3:30pm-4:30pm
TA	Amir Azarmehr (Office hours: TBD)
Prerequisites	This is a rapid course covering advanced algorithms. It is intended primarily for PhD students in Khoury, but if you are not one, you need the instructor's permission to enroll. You will be well-prepared for this course if you have completed an undergraduate algorithms course (e.g. CS3000) and are comfortable in mathematical reasoning and communication. The ability to reason mathematically is more important than prior knowledge. Given that this is a PhD-level course, participants may have varying backgrounds, and you might need to address some gaps in your knowledge yourself. If you have any concerns about your background, I encourage you to discuss them with me during the first week of classes.

Course Overview

We will cover the (mathematical) foundations of algorithms. We will review some material covered in CS5800/CS3000 and then cover more advanced algorithms.

Grading

- 20% Midterm 1
- 20% Midterm 2
- 30% Final Exam
- 30% Homework Assignments (expect about 4 HWs)

Schedule

#	WD	Date	Topics	References	Notes
1	Tu	1/7	Introduction		

Stable Matching

Stable Matching

- Statement of the problem
  - Two sides of the market to be matched.
  - Participants on both sides care about to whom they are matched.
  - Money can’t be used to determine the assignment.

- Examples
  - Marriage & dating markets
  - School choice programs
  - College admissions
  - Medical residencies
  - Job assignments in firms
  - Kidney exchange

Example

- Input: n doctors, n hospitals, each with a list of preferences.

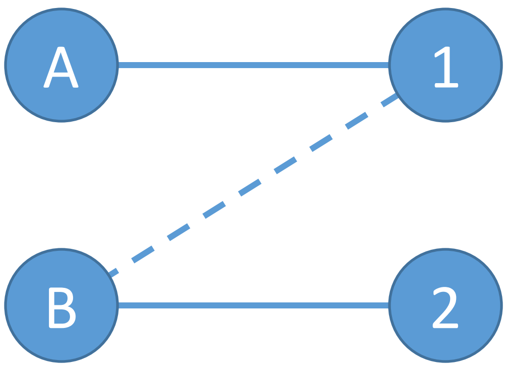
	1st	2nd	3rd	4th	5th
MGH	Bob	Alice	Dorit	Ernie	Clara
BW	Dorit	Bob	Alice	Clara	Ernie
BID	Bob	Ernie	Clara	Dorit	Alice
MTA	Alice	Dorit	Clara	Bob	Ernie
CH	Bob	Dorit	Alice	Ernie	Clara

	1st	2nd	3rd	4th	5th
Alice	CH	MGH	BW	MTA	BID
Bob	BID	BW	MTA	MGH	CH
Clara	BW	BID	MTA	CH	MGH
Dorit	MGH	CH	MTA	BID	BW
Ernie	MTA	BW	CH	BID	MGH

- **Matching:** Pair doctors/hospitals such that no doctor or hospital is matched more than once.
- **Perfect matching:** every doctor gets a job.
- **Example perfect matching:**
  - {(Alice, MTA), (Bob, BID), (Clara, CH), (Dorit, MGH), (Ernie, BW)}

Stable Matching

- Idea: want to pair up members of two sets according to their preferences
- Can't hope to give everyone their first choice!
- Instead, prevent any *instabilities*:



B prefers 1 to 2
and
1 prefers B to A

Ask the Audience

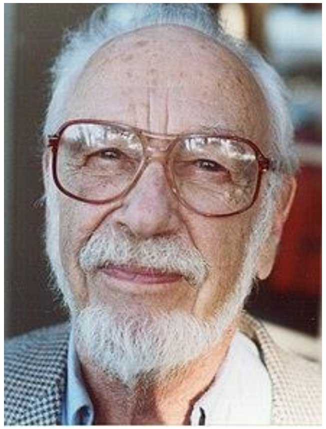
- Either find a stable matching or convince yourself that such a matching doesn't exist.

	1st	2nd	3rd
MGH	Alice	Bob	Clara
BW	Bob	Clara	Alice
BID	Alice	Clara	Bob

	1st	2nd	3rd
Alice	BW	BID	MGH
Bob	BW	MGH	BID
Clara	MGH	BID	BW

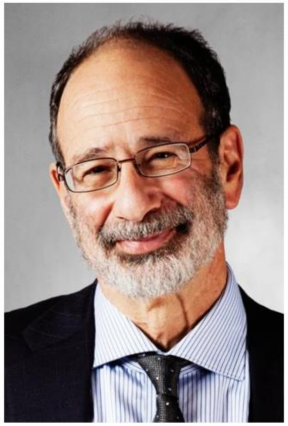
Gale-Shapley Algorithm

- The Gale-Shapley algorithm finds a stable matching.
- It was designed by David Gale and Lloyd Shapley in 1962.
- National system for matching US medical school graduates to medical residencies:
  - Roughly 40,000 doctors per year
  - Assignment is almost entirely algorithmic



David Gale

Lloyd Shapley



Alvin Roth

Gale-Shapley Algorithm

- Input: Preferences for n doctors and n hospitals.

- $M \leftarrow \emptyset$
- While (there is an unmatched hospital h):
  - h "offers" to their favorite doctor d to whom they have not made an offer yet.
  - **If** d has no job then add (h, d) to M.
  - **Else** let $(h', d) \in M$ be the current job for d:
    - If d prefers h' over h:
      - Do nothing.
    - Else:
      - Remove (h', d) from M, and instead add (h, d) to M.
- Return M.

Gale-Shapley Demo

	1st	2nd	3rd	4th	5th
MGH	Bob	Alice	Dorit	Ernie	Clara
BW	Dorit	Bob	Alice	Clara	Ernie
BID	Bob	Ernie	Clara	Dorit	Alice
MTA	Alice	Dorit	Clara	Bob	Ernie
CH	Bob	Dorit	Alice	Ernie	Clara

	1st	2nd	3rd	4th	5th
Alice	CH	MGH	BW	MTA	BID
Bob	BID	BW	MTA	MGH	CH
Clara	BW	BID	MTA	CH	MGH
Dorit	MGH	CH	MTA	BID	BW
Ernie	MTA	BW	CH	BID	MGH

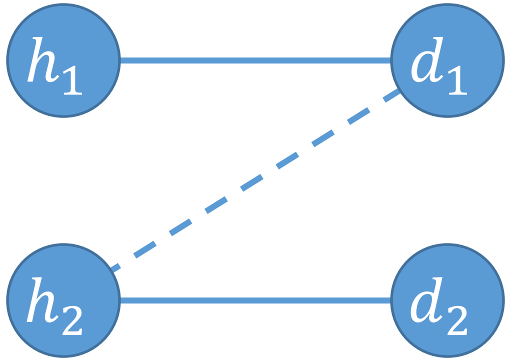
Observations

- The algorithm has to terminate.
- Any doctor that gets an offer will always hold an offer.
- “Hospitals gradually go down”
- “Doctors gradually go up”

Optimality

- Why is the solution M a stable matching?

- Suppose there is an instability:



- Case 1: h_2 never offered to d_1

- Case 2: h_2 made on offer to d_1

Real-world Impact

TABLE I

STABLE AND UNSTABLE (CENTRALIZED) MECHANISMS

Market	Stable	Still in use (halted unraveling)
American medical markets		
NRMP	yes	yes (new design in ’98)
Medical Specialties	yes	yes (about 30 markets)
British Regional Medical Markets		
Edinburgh (‘69)	yes	yes
Cardiff	yes	yes
Birmingham	no	no
Edinburgh (‘67)	no	no
Newcastle	no	no
Sheffield	no	no
Cambridge	no	yes
London Hospital	no	yes
Other healthcare markets		
Dental Residencies	yes	yes
Osteopaths (<‘94)	no	no
Osteopaths (≥‘94)	yes	yes
Pharmacists	yes	yes
Other markets and matching processes		
Canadian Lawyers	yes	yes (except in British Columbia since 1996)
Sororities	yes (at equilibrium)	yes

Table 1. Reproduced from Roth (2002, Table 1).

Shapley and Roth won the 2012 Nobel Prize for their work on stable matchings