# Advanced Algorithms

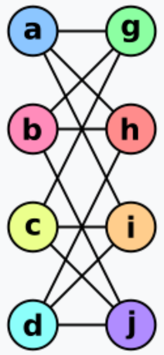
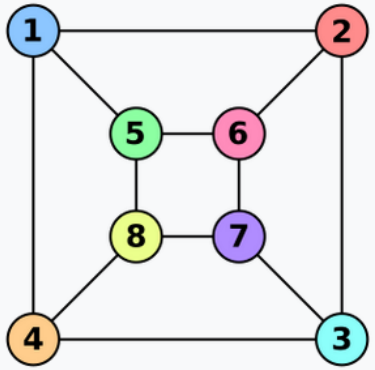
August 28th, 2025

# Feelings about the Exercise Set?

- Asymptotics, graphs and NP-hardness. Rules of logarithms.
- I have office hours after this
- Submit next Tuesday in class
  - latex or handwritten

# Graph Isomorphism

- Check if two graphs are "the same"
- The problem is in NP
- Unknown whether the problem is NP-hard
- Babai 2015 gave an $O\left(n^{\log n}\right)$ time algorithm. Retracted then republished.
- True complexity remains unclear . . .

| Graph G | Graph H | An isomorphism between G and H |
|---|---|---|
|  |  | *f*(*a*) = 1<br>*f*(*b*) = 6<br>*f*(*c*) = 8<br>*f*(*d*) = 3<br>*f*(*g*) = 5<br>*f*(*h*) = 2<br>*f*(*i*) = 4<br>*f*(*j*) = 7 |

# Logistics

- [http://www.cs.pomona.edu/classes/cs181aa/](http://www.cs.pomona.edu/classes/cs181aa/)

My office is Edmunds 223

<u>Office hours:</u>

Monday 10:00 – 11:00am

**Thursdays 2:40 – 4:00pm**

By appointment

If I'm in my office, feel free to knock.

**CS 181: Advanced Algorithms - Fall 2025**

Home Syllabus Schedule Assignments Resources

## Announcements

- Welcome to CS 181: Advanced Algorithms! Read the syllabus and check the schedule.

## Syllabus

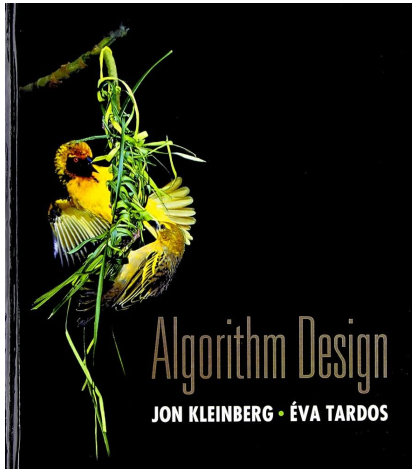
**Logistics:** The instructor for this course is me, Professor Zlatin. We meet on Tuesdays and Thursdays from 1:15 - 2:30pm in Estella 1249. My office hours are on Mondays from 10 - 11am, Thursdays 2:40 - 4:00pm, and by appointment in Edmunds 223. I am happy to talk about the class, CS theory or whatever is on your mind. The best way to reach me is by email. There will also be a course Slack channel.

**Course Description:** Did you enjoy your first algorithms course and want to go deeper? Then this course is for you! We will cover a range of topics which have become fundamental tools in the design of modern algorithms. We begin by studying widely applicable network optimization problems. We will then go beyond the realm of exact algorithms and learn how to approach problems which are known to be intractable, or for which we only have partial information. The class will culminate in a course project.

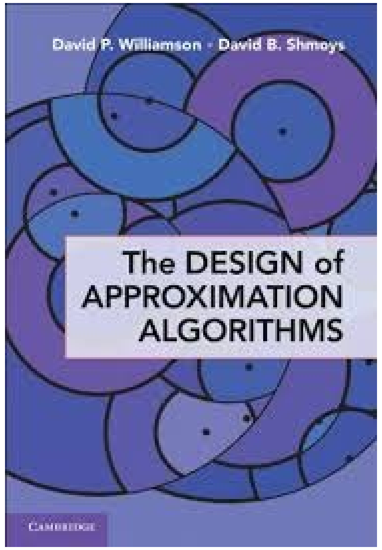
# Complete tentative Schedule

| Date | Topic | Readings | Due |
|---|---|---|---|
| **Part 1 · Network Optimization** | | | |
| 08/26 | Welcome to the course | slides | Welcome survey |
| 08/28 | Ford-Fulkerson, max-flow min-cut theorem | | |
| 09/02 | Edmonds-Karp: a polynomial algorithm for max-flow | | Exercise set 0 |
| 09/04 | Push-relabel: a non-augmenting-path algorithm for max-flow | | |
| 09/09 | Image segmentation, Bipartite Matching | | |
| 09/11 | Hall's Theorem, The Hungarian Algorithm | | |
| 09/16 | Generalizations: min-cost flow and non-bipartite matchings | | |
| **Part 2 · Combinatorial Optimization** | | | |
| 09/18 | Intro to Linear Programming | | |
| 09/23 | Geometry and modeling with LPs | | |
| 09/25 | Deriving the dual, duality of max-flow and min-cut | | |
| 09/30 | Integer programming, integrality gaps | | |
| 10/02 | The bipartite perfect matching polytope | | |
| 10/07 | Edmonds' blossom polytope / Konig's Theorem / By request | | |
| **Part 3 · Dealing with Intractability** | | | |
| 10/09 | Intro to approximation algos, knapsack, scheduling | | |
| 10/14 | Fall break | | |
| 10/16 | No class | | |
| 10/21 | Maximum Cut, Algorithms for k-center | | |
| 10/23 | Network Design: Steiner trees | | |
| 10/28 | The Traveling Salesperson Problem | | |
| 10/30 | Set Cover | | |
| 11/04 | Parameterized algorithms for vertex cover | | |
| **Part 4 · Dealing with Uncertainty** | | | |
| 11/06 | Intro to Online algorithms: ski-rentals and secretaries | | |
| 11/11 | Online scheduling, steiner tree | | |
| 11/13 | Caching / paging | | |
| 11/18 | Online Bipartite Matching | | |
| 11/20 | Intro to streaming | | |
| 11/25 | By request | | |
| 12/02 | By request | | |
| **Final Project Presentations** | | | |
| 12/09 | In class 2:00 - 5:00pm | | |

# Resources




- Algorithm Design by Kleinberg and Tardos
  → Especially useful for the first part of the course
  → Copies in the Edmunds computer lab




- A Second Course in Algorithms by Tim Roughgarden
  → This is an online course
  → Great notes, videos on youtube.

- Design of Approximation Algorithms by Williamson and Shmoys
  → Best book on approx algos I know, copies in lab

# Workload

- Come to class and engage

- Exercise sets most weeks → due in class one week later
  - This week's is unusually long (sorry)

- Four assignments:
  - Test the core skills I want **each student** to take away from this course
  - These are harder, collaboration is strongly encouraged!
  - Write up solutions yourself though

- Final Project*

# AI policy

Basically, don't use it on the assignments

May use to help you learn:

- Ask questions, explain concepts, generate examples, quiz yourself, help with the exercise sets (must write solutions yourself)
- You may not use AI tools for assistance on assignments.

- I genuinely believe that this is an important piece of the learning process
- Always feel free to come to me for help on assignments, I can give hints or guide your thought process

# Slack

- You should be in it
- If not, come see me or email me

Thank you for filling out the course survey

# Stable Matchings

# Activity

1. Divide yourselves into two equal groups

2. Consider the preferences on your sheet

3. I will set a 1:00 minute timer

Your goal: pair with a partner highest on your preference list

# Matching med-school students to hospitals

- Goal. Given a set of preferences among hospitals and med-school students, design a self-reinforcing admissions process.

- Unstable pair. Hospital $h$ and student $s$ form an unstable pair if both:
  - $h$ prefers $s$ to one of its admitted students.
  - $s$ prefers $h$ to assigned hospital.

- Stable assignment. Assignment with no unstable pairs.
  - Self-interest prevents any hospital–student side deal.

# Stable matching problem: the input

Input: A set of $n$ hospitals $H$ and a set of $n$ students $S$.

- Each hospital $h \in H$ ranks students.
- Each student $s \in S$ ranks hospitals.

one student per hospital (for now)

| | 1st (favorite) | 2nd | 3rd (least favorite) |
|---|---|---|---|
| Atlanta | Xavier | Yolanda | Zeus |
| Boston | Yolanda | Xavier | Zeus |
| Chicago | Xavier | Yolanda | Zeus |

hospitals' preference lists

| | 1st (favorite) | 2nd | 3rd (least favorite) |
|---|---|---|---|
| Xavier | Boston | Atlanta | Chicago |
| Yolanda | Atlanta | Boston | Chicago |
| Zeus | Atlanta | Boston | Chicago |

students' preference lists

From slides on Algorithm Design

# Perfect Matchings

Definition: A matching $M$ is a set of pairs $(h, s)$ with $h \in H$ and $s \in S$ such that:

- Each hospital $h \in H$ appears in at most one pair of $M$.
- Each student $s \in S$ appears in at most one pair of $M$.

Definition: A matching $M$ is perfect if $|M| = |H| = |S| = n$.

| | 1st | 2nd | 3rd |
|---|---|---|---|
| **Atlanta** | Xavier | Yolanda | Zeus |
| **Boston** | Yolanda | Xavier | Zeus |
| **Chicago** | Xavier | Yolanda | Zeus |

| | 1st | 2nd | 3rd |
|---|---|---|---|
| **Xavier** | Boston | Atlanta | Chicago |
| **Yolanda** | Atlanta | Boston | Chicago |
| **Zeus** | Atlanta | Boston | Chicago |

**a perfect matching M = { A–Z, B–Y, C–X }**

# Instability in matchings

Def. Given perfect matching $M$, hospital $h$ and student $s$ are an unstable pair if:

- $h$ prefers $s$ to matched student.
- $s$ prefers $h$ to matched hospital

Key point. An unstable pair $h$– $s$ could each improve by joint action.

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Atlanta | Xavier | Yolanda | Zeus |
| Boston | Yolanda | Xavier | Zeus |
| Chicago | Xavier | Yolanda | Zeus |

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Xavier | Boston | Atlanta | Chicago |
| Yolanda | Atlanta | Boston | Chicago |
| Zeus | Atlanta | Boston | Chicago |

A–Y is an unstable pair for matching M = { A–Z, B–Y, C–X }

# Pop Quiz

Which pair is unstable in the matching { A–X, B–Z, C–Y } ?

1. A–Y.
2. B–X.
3. B–Z.
4. None of the above.

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Atlanta | Xavier | Yolanda | Zeus |
| Boston | Yolanda | Xavier | Zeus |
| Chicago | Xavier | Yolanda | Zeus |

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Xavier | Boston | Atlanta | Chicago |
| Yolanda | Atlanta | Boston | Chicago |
| Zeus | Atlanta | Boston | Chicago |

# Pop Quiz

Notice, an unstable pair is *not currently matched*

Which pair is unstable in the matching { A–X, B–Z, C–Y } ?

1. A–Y.
2. B–X.
3. B–Z.
4. None of the above.

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Atlanta | Xavier | Yolanda | Zeus |
| Boston | Yolanda | Xavier | Zeus |
| Chicago | Xavier | Yolanda | Zeus |

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Xavier | Boston | Atlanta | Chicago |
| Yolanda | Atlanta | Boston | Chicago |
| Zeus | Atlanta | Boston | Chicago |

# Stable matching problem

Def. A stable matching is a perfect matching with no unstable pairs.

Stable matching problem. Given the preference lists of $n$ hospitals and $n$ students, find a stable matching (if one exists).

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Atlanta | Xavier | Yolanda | Zeus |
| Boston | Yolanda | Xavier | Zeus |
| Chicago | Xavier | Yolanda | Zeus |

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Xavier | Boston | Atlanta | Chicago |
| Yolanda | Atlanta | Boston | Chicago |
| Zeus | Atlanta | Boston | Chicago |

**a stable matching M = { A–X, B–Y, C–Z }**

# Stable roommate problem

Question: Do stable matchings always exist?

Answer: Not so obvious . . .

Stable roommate problem:

- $2n$ people; each person ranks others from 1 to $2n - 1$.
- Assign roommate pairs so that no unstable pairs exist.

# Stable roommate problem

Question: Do stable matchings always exist?

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

no perfect matching is stable

$A–B,\ C–D \Rightarrow B–C$ unstable

$A–C,\ B–D \Rightarrow A–B$ unstable

$A–D,\ B–C \Rightarrow A–C$ unstable

- Observation. Stable matchings **need not exist** in the stable roommate problem

# Gale-Shapley Algorithm

1. Each student, propose a match to your highest ranked hospital.

2. Each hospital **tentatively** accept their most preferred student who proposed to them.

3. Unmatched students apply to the **next highest** ranked hospital in their list.

4. Again, each hospital **tentatively** matches with their most preferred student who has proposed

5. Repeat until everyone is matched. Done!

# Observations

1. **Monotonicity for hospitals:** A hospital's tentative match can only get better over time.

2. **No re-proposals:** Student never return to a hospital they've already proposed to.

3. **Runtime:** Point number 2 implies that the process ends after at most ______ rounds.

   There are at most _______ proposals per round.

   Thus, there at most _______ proposals total.

# Is this a stable matching?

- Take any student $s$ and hospital $h$ who are not matched to each other.
- We want to argue that they don't prefer one another to their partners

Key question: did $s$ ever propose to $h$?

- If not, then $s$ must have been matched earlier in their preference list.
- If so, then $h$ must have either rejected $s$, or later found someone better to replace $s$.

# Conclusion

Stable matching problem. Given $n$ hospitals and $n$ students, and their preference lists, find a stable matching if one exists

Theorem [Gale-Shapley 1962]: A stable matching always exists, and it can be computed in $O(n^2)$ time.

**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. Accordingly, in order for a college to receive $q$ acceptances, it will generally have to offer to admit more than $q$ applicants. The problem of determining how many and which ones to admit requires some rather involved guesswork. It may not be known (a) whether a given applicant has also applied elsewhere; if this is known it may not be known (b) how he ranks the colleges to which he has applied; even if this is known it will not be known (c) which of the other colleges will offer to admit him. A result of all this uncertainty is that colleges can expect only that the entering class will come reasonably close in numbers to the desired quota, and be reasonably close to the attainable optimum in quality.

# Some extensions

student unwilling to work in Cleveland

- Extension 1. Some agents declare others as unacceptable.
- Extension 2. Some hospitals have more than one position
- Extension 3. Unequal number of positions and students

$\geq$ 43K med-school students, only 31K positions

Def. Matching $M$ is unstable if there is a hospital $h$ and student $s$ such that:

- $h$ and $s$ are acceptable to each other, and
- Either $s$ is unmatched, or $s$ prefers $h$ to assigned hospital; and
- Either $h$ does not have all its places filled, or $h$ prefers $s$ to at least one of its assigned students.

Theorem. There exists a stable matching. Simple adaptation of Gale-Shapley.

# Students vs Hospitals

- What if we switch the roles of hospitals and students?
  - Hospitals propose in each round.

- Does this still always yield a stable matching?

- Which stable matching is preferred?

☆ If students propose, the stable matching is **student-optimal.** They get the best hospital possible in any stable matching.

If hospitals propose, the stable matching is **hospital-optimal.**

# This is really used


**The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design**

By Alvin E. Roth and Elliott Peranson*

*We report on the design of the new clearinghouse adopted by the National Resident Matching Program, which annually fills approximately 20,000 jobs for new physicians. Because the market has complementarities between applicants and between positions, the theory of simple matching markets does not apply directly. However, computational experiments show the theory provides good approximations. Furthermore, the set of stable matchings, and the opportunities for strategic manipulation, are surprisingly small. A new kind of "core convergence" result explains this; that each applicant interviews only a small fraction of available positions is important. We also describe engineering aspects of the design process.* (JEL C78, B41, J44)

THE MATCH
NATIONAL RESIDENT MATCHING PROGRAM

**How Game Theory Helped Improve New York City's High School Application Process**

By TRACY TULLIS DEC. 5, 2014

Tuesday was the deadline for eighth graders in New York City to submit applications to secure a spot at one of 426 public high schools. After months of school tours and tests, auditions and interviews, 75,000 students have entrusted their choices to a computer program that will arrange their school assignments for the coming year. The weeks of research and deliberation will be reduced to a fraction of a second of mathematical calculation: In just a couple of hours, all the sorting for the Class of 2019 will be finished.

☆ Robust to preference falsification

# A Question for Thought

- What if we have a **one-sided** matching market?

- Imagine a collection of $n$ items, where each of $n$ people has preferences over these them. However, the items have no preferences over people.

- Can you think of an algorithm to allocate items to people in a principled way? What properties would we like the allocation to satisfy?