Objectives

- Introduction to problem solving
  - Our process, through an example
- Wiki
  - Everyone log in okay?
  - Decide on either using a blog or wiki?

Matching Residents to Hospitals

- **Goal**: Given a set of preferences among hospitals and medical school students, design a *self-reinforcing* admissions process.
- Applicant *x* and hospital *y* are ***unstable*** if
  - *x* prefers *y* to its assigned hospital
  - *y* prefers *x* to one of its admitted students
- **Stable assignment**: Assignment with no unstable pairs
  - No incentive for some pair of participants to undermine assignment by joint action
    - Unstable pair could each improve by swapping with current assignment
    - (Self-reinforcing part)

What details make this problem tricky?
What info do we need to solve problem?

Stable Matching Problem

Simplified version of resident-matching problem

- **Goal**: Given *n* men and *n* women, find a "suitable" matching
  - Participants rank members of opposite sex
  - Each man lists women in order of preference from best to worst
  - Each woman lists men in order of preference from best to worst

favorite → least favorite

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite → least favorite

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Stable Matching Goals

- **Perfect matching**: everyone is matched monogamously
  - Each man is paired with exactly one woman
  - Each woman is paired with exactly one man
- **Stability**: no incentive for some pair of participants to undermine assignment by joint action
  - An *unmatched* pair *m-w* is **unstable** if man *m* and woman *w* prefer each other to current partners
  - Unstable pair *m-w* could each improve by eloping
- **Stable matching**: perfect matching with no unstable pairs

Stable matching problem:
Given the preference lists of *n* men and *n* women, find a stable matching if one exists.

Analyzing Stability

- Is pairing X-C, Y-B, Z-A stable?
  - Recall defn. Instable: *m* prefers *w* to his woman; *w* prefers *m* to her man

favorite → least favorite

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite → least favorite

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Analyzing Stability

- Is pairing X-C, Y-B, Z-A stable?
- No. Bertha and Xavier prefer each other

favorite → least favorite

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite → least favorite

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Stable Matching Problem

- Is pairing X-A, Y-B, Z-C stable?
- Yes.

favorite → least favorite

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite → least favorite

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Any Questions?

- What are you wondering about this problem/ its solution at this point?

Any Questions?

- What are you wondering about this problem/ its solution at this point?
- Hopefully:
  - Is there a stable matching for every pair of preference lists?
  - If so, is there an algorithm to find the stable matching?
  - Can we be fair in the matching? (preferences)
  - Will the matching always be the same?

Thoughts on Solving Problem

- What do we need to solve the problem?
- What do we know?
- Where should the state start?
- What are some initial ideas about approaches?

Thoughts on Solving Problem

- Initially, no one is matched
- Pick an arbitrary man and have him match with his favorite woman.
  - Are we guaranteed that pair will be part of a stable matching?
- Should a woman accept her first offer? If not, what should she do?
- When are we done? Do we need to consider all combinations?

Propose-And-Reject Algorithm

[Gale-Shapley 1962]

- Intuitive method that guarantees finding a stable matching

```
Initialize each person to be free
while (some man is free and hasn't proposed to every woman)
    Choose such a man m
    w = 1st woman on m's list to whom m has not yet proposed
    if (w is free)
        assign m and w to be engaged
    else if (w prefers m to her fiancé m')
        assign m and w to be engaged and m' to be free
    else
        w rejects m
```

Applying the Algorithm

favorite → 1st, 2nd, 3rd ← least favorite

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

```
Initialize each person to be free
while (some man is free and hasn't proposed to every woman)
    Choose such a man m
    w = 1st woman on m's list to whom m has not yet proposed
    if (w is free)
        assign m and w to be engaged
    else if (w prefers m to her fiancé m')
        assign m and w to be engaged and m' to be free
    else
        w rejects m
```

Observations about the Algorithm

- What can we say about any woman's partner during the execution of the algorithm?
- How does a woman's state change over the execution of the algorithm?
- What can we say about a man's partner?

Observations about the Algorithm

- What can we say about any woman's partner during the execution of the algorithm?
  - Observation 1. He gets "better" → she prefers him over her last partner
- How does a woman's state change over the execution of the algorithm?
  - Observation 2. Once a woman is matched, she never becomes unmatched; she only "trades up"
- What can we say about a man's partner?
  - Observation 3. She gets "worse"

Proving Correctness

- Need to show
  - Algorithm terminates
  - Result is a perfect matching
  - Result is a stable matching

Proof of Correctness: Termination

- Claim. Algorithm terminates after at most n^2 iterations of while loop.
  - Hint: How wouldn't the algorithm terminate?

Proof of Correctness: Termination

- Claim. Algorithm terminates after at most n^2 iterations of while loop.
- Pf. Each time through the while loop, a man proposes to a new woman. There are only n^2 possible proposals.

Tighter: n(n-1) + 1 proposals required

Algorithm Analysis

Prove that final matching is a *perfect* matching

- Hint: in algorithm, we know if *m* is free at some point in the execution of the algorithm, then there is a woman to whom he has not yet proposed.

Proof of Correctness: Perfection

- Claim. All men and women get matched.
- Pf. (by contradiction)
  - Where should we start?

Suppose that some man *m* is not matched upon termination of algorithm

Proof of Correctness: Perfection

- Claim. All men and women get matched.
- Pf. (by contradiction)
  - Suppose that *m* is not matched upon termination of algorithm
  - Then some woman, say *w*, is not matched upon termination.
  - By Observation 2, *w* was never proposed to.
  - But, last man proposed to everyone, since he ends up unmatched
    - (by the while loop's condition)
  - Contradiction ▪

Proof of Correctness: Stability

- Claim. No unstable pairs.

What does it mean to be unstable, given matching S*?

S*
Amy-Yancey
Bertha-Zeus
…

How do you think we should approach this proof?

Proof of Correctness: Stability

S*
Amy-Yancey
Bertha-Zeus
…

- Claim. No unstable pairs.
- Pf. (by contradiction)
  - Suppose A-Z is an unstable pair: each prefers each other to partner in Gale-Shapley matching S*.

How could that happen?
What are the possibilities that lead to this?

Proof of Correctness: Stability

S*
Amy-Yancey
Bertha-Zeus
…

- Claim. No unstable pairs.
- Pf. (by contradiction)
  - Suppose A-Z is an unstable pair: each prefers each other to partner in Gale-Shapley matching S*.
  - Case 1: Z never proposed to A.
    - ⇒ Z prefers his GS partner to A.
    - ⇒ A-Z is stable.

    ← men propose in decreasing order of preference
  - Case 2: Z proposed to A.
    - ⇒ A rejected Z (right away or later)
    - ⇒ A prefers her GS partner to Z.
    - ⇒ A-Z is stable.

    ← women only trade up
  - In either case A-Z is stable, a contradiction. ▪

Summary So Far…

- **Stable matching problem.** Given *n* men and *n* women and their preferences, find a stable matching if one exists.
- **Gale-Shapley algorithm.** Guarantees to find a stable matching for *any* input

Remaining Questions:
- If there are multiple stable matchings, which one does GS find? *(see book)*
- How to implement GS algorithm efficiently? (Monday)
  - What is our goal running time?

Review: Our Process

1. Understand/identify problem
   - Simplify as appropriate
2. Design a solution
3. Analyze
   - Correctness, efficiency
   - May need to go back to step 2 and try again
4. Implement
   - Within bounds shown in analysis

Stable Matching Summary

- **Stable matching problem.** Given preference profiles of *n* men and *n* women, find a *stable* matching.

  no man and woman prefer to be with each other than assigned partner

- **Gale-Shapley algorithm.** Finds a stable matching in $O(n^2)$ time.
  - Claim: can implement algorithm *efficiently*

Assignments

- Review Chapter 1
- Read Chapter 2
- Journal due next Wednesday
  - Preface
  - Chapter 1
  - Beginning of Chapter 2

Journals about Text Book

- New entry for each week OR chapter OR section
- Include a page on the Preface too (up to Overview)
- Clearly delineate sections/chapters
- **What to Write in Your Notes**
  - Brief summary of what the chapter/section covers (~1 paragraph of about 5 sentences/section; feel free to write more if that will help you)
  - Include motivations for the given problem, as appropriate
  - Questions you have about motivation/solution/proofs/analysis
  - Discuss anything that makes more sense after reading it again, after it was presented in class (or vice versa)
  - Anything that you want to remember, anything that will help you
  - Say something about how readable/interesting the section was on scale of 1 to 10