CSE 417: Winter '10
Algorithms and Computational Complexity

Elisa Celis - CSE 210
ecelis @ cs.washington.edu

http://cse417win10.wordpress.com

Original Slides by:
Kevin Wayne
Princeton University

Administrative Stuff

Lectures. Elisa Celis
• MF 2:30 - 3:50, Low 201.
• Attendance is expected.
• MF 11-12 and by appointment, CSE 210

TAs. Emily Wang, Saptarshi Bhattacharya
• TWR 2:30-3:30pm, CSE 218.

Prereq. CSE 142, 143 and 373, or instructor's permission.
Familiarity with basic programming and mathematical proofs strongly suggested.

Enrollment. Class is officially full. See me after class for permission to enroll.

Textbook. Algorithm Design by Éva Tardos and Jon Kleinberg.
Grades

**Grading.**
- 60-65% Weekly homework assignments, due Friday 2:30pm in class.
  - One programming assignment.
  - Some extra credit.
- 15-20% In-class midterm (Monday Feb 8th)
- 20-25% Take-home final.
- Class participation for borderline cases.

**Homework.**
- There is homework due this Friday.
- See handout for writing solutions.
- You get one free "late day". (see syllabus; ask if unsure)

Collaboration

**Collaboration policy.** (see syllabus; ask if unsure)
- Course materials are always permitted.
- You are encouraged to attend office hours as needed.
- No external resources, e.g., Google.

**Peer collaboration encouraged, BUT:**
- You must write up solutions individually.
- Cite any and all collaborators.
- Exceptions are: exams, extra credit problems.
<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Algorithmic Paradigms</th>
</tr>
</thead>
</table>
| [Knuth, TAOCP] An algorithm is a finite, definite, effective procedure, with some input and some output. | Design and analysis of computer algorithms.  
- Runtime Analysis.  
- Greedy Algorithms.  
- Dynamic programming.  
- Divide-and-conquer.  
- Reductions and Intractability.  
- Coping with intractability.  
Critical thinking and problem-solving. |

"Great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing."  
- Francis Sullivan
Applications

Wide range of applications.

- Game Theory.
- Databases.
- Social Networks.
- Data analysis.
- Signal processing.
- Economics.
- Operations research.
- Artificial intelligence.
- Computational biology.
- Search (e.g. Bing, Google)
- Ebay
- ...

We focus on algorithms and techniques that are useful in practice.
1.1 A First Problem: Stable Matching

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

Unstable pair: 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.
- Natural and desirable condition.
- Individual self-interest will prevent any applicant/hospital deal from being made.
Stable Matching Problem

**Goal.** Given \( n \) men and \( n \) women, find a "suitable" matching.
- Participants rate 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.

<table>
<thead>
<tr>
<th>Favorite</th>
<th>Least Favorite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xavier</td>
<td>Amy</td>
</tr>
<tr>
<td>Bertha</td>
<td>Amy</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha</td>
</tr>
</tbody>
</table>

*Men's Preference Profile*

<table>
<thead>
<tr>
<th>Favorite</th>
<th>Least Favorite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Yancey</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
</tr>
<tr>
<td>Clare</td>
<td>Xavier</td>
</tr>
</tbody>
</table>

*Women's Preference Profile*

**Perfect matching:** everyone is matched monogamously.
- Each man gets exactly one woman.
- Each woman gets exactly one man.

**Stability:** no incentive for some pair of participants to undermine assignment by joint action.
- In matching \( M \), 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.
Stable Matching Problem

Q. Is assignment X-C, Y-B, Z-A stable?

<table>
<thead>
<tr>
<th>Men's Preference Profile</th>
<th>Women's Preference Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>favorite</strong></td>
<td><strong>least favorite</strong></td>
</tr>
<tr>
<td>1st</td>
<td>2nd</td>
</tr>
<tr>
<td>Xavier</td>
<td>Amy</td>
</tr>
<tr>
<td>Yancey</td>
<td>Bertha</td>
</tr>
<tr>
<td>Zeus</td>
<td>Amy</td>
</tr>
</tbody>
</table>

A. No. Bertha and Xavier will hook up.
Stable Matching Problem

Q. Is assignment X-A, Y-B, Z-C stable?

A. Yes.

<table>
<thead>
<tr>
<th>Men’s Preference Profile</th>
<th>Women’s Preference Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>favorite</strong></td>
<td><strong>least favorite</strong></td>
</tr>
<tr>
<td>Xavier</td>
<td>Amy</td>
</tr>
<tr>
<td>Yancey</td>
<td>Bertha</td>
</tr>
<tr>
<td>Zeus</td>
<td>Amy</td>
</tr>
</tbody>
</table>
Propose-And-Reject Algorithm


```
Propose-and-reject algorithm.
Initialize each person to be free.
while (some free man has not proposed to every woman) {
    Choose such a man m
    w = best woman on m's list who has not rejected him
    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
}
```

### Men's Preference Profile

<table>
<thead>
<tr>
<th>Man</th>
<th>Amy</th>
<th>Diane</th>
<th>Erika</th>
<th>Clare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victor</td>
<td>Bertha</td>
<td>Amy</td>
<td>Diane</td>
<td>Erika</td>
</tr>
<tr>
<td>Wyatt</td>
<td>Diane</td>
<td>Bertha</td>
<td>Amy</td>
<td>Clare</td>
</tr>
<tr>
<td>Xavier</td>
<td>Bertha</td>
<td>Erika</td>
<td>Clare</td>
<td>Diane</td>
</tr>
<tr>
<td>Yancey</td>
<td>Amy</td>
<td>Diane</td>
<td>Clare</td>
<td>Bertha</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha</td>
<td>Diane</td>
<td>Amy</td>
<td>Erika</td>
</tr>
</tbody>
</table>

### Women's Preference Profile

<table>
<thead>
<tr>
<th>Woman</th>
<th>Zeus</th>
<th>Victor</th>
<th>Wyatt</th>
<th>Yancey</th>
<th>Xavier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Zeus</td>
<td>Victor</td>
<td>Wyatt</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Wyatt</td>
<td>Yancey</td>
<td>Victor</td>
<td>Zeus</td>
</tr>
<tr>
<td>Clare</td>
<td>Wyatt</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
<td>Victor</td>
</tr>
<tr>
<td>Diane</td>
<td>Victor</td>
<td>Zeus</td>
<td>Yancey</td>
<td>Xavier</td>
<td>Wyatt</td>
</tr>
<tr>
<td>Erika</td>
<td>Yancey</td>
<td>Wyatt</td>
<td>Zeus</td>
<td>Xavier</td>
<td>Victor</td>
</tr>
</tbody>
</table>
### Men's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victor</td>
<td>Bertha</td>
<td>Amy</td>
<td>Diane</td>
<td>Erika</td>
</tr>
<tr>
<td>Wyatt</td>
<td>Diane</td>
<td>Bertha</td>
<td>Amy</td>
<td>Clare</td>
</tr>
<tr>
<td>Xavier</td>
<td>Bertha</td>
<td>Erika</td>
<td>Clare</td>
<td>Diane</td>
</tr>
<tr>
<td>Yancey</td>
<td>Amy</td>
<td>Diane</td>
<td>Clare</td>
<td>Bertha</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha</td>
<td>Diane</td>
<td>Amy</td>
<td>Erika</td>
</tr>
</tbody>
</table>

Victor proposes to Bertha.

### Women's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Zeus</td>
<td>Victor</td>
<td>Wyatt</td>
<td>Yancey</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Wyatt</td>
<td>Yancey</td>
<td>Victor</td>
</tr>
<tr>
<td>Clare</td>
<td>Wyatt</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Diane</td>
<td>Victor</td>
<td>Zeus</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Erika</td>
<td>Yancey</td>
<td>Wyatt</td>
<td>Zeus</td>
<td>Xavier</td>
</tr>
</tbody>
</table>

Victor proposes to Bertha.
- Bertha accepts since previously unmatched.

### Men's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victor</td>
<td>Bertha</td>
<td>Amy</td>
<td>Diane</td>
<td>Erika</td>
</tr>
<tr>
<td>Wyatt</td>
<td>Diane</td>
<td>Bertha</td>
<td>Amy</td>
<td>Clare</td>
</tr>
<tr>
<td>Xavier</td>
<td>Bertha</td>
<td>Erika</td>
<td>Clare</td>
<td>Diane</td>
</tr>
<tr>
<td>Yancey</td>
<td>Amy</td>
<td>Diane</td>
<td>Clare</td>
<td>Bertha</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha</td>
<td>Diane</td>
<td>Amy</td>
<td>Erika</td>
</tr>
</tbody>
</table>

Victor proposes to Bertha.

### Women's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Zeus</td>
<td>Victor</td>
<td>Wyatt</td>
<td>Yancey</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Wyatt</td>
<td>Yancey</td>
<td>Victor</td>
</tr>
<tr>
<td>Clare</td>
<td>Wyatt</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Diane</td>
<td>Victor</td>
<td>Zeus</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Erika</td>
<td>Yancey</td>
<td>Wyatt</td>
<td>Zeus</td>
<td>Xavier</td>
</tr>
</tbody>
</table>
### Men's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victor</td>
<td>Bertha</td>
<td>Amy</td>
<td>Diane</td>
<td>Erika</td>
</tr>
<tr>
<td>Wyatt</td>
<td>Diane</td>
<td>Bertha</td>
<td>Amy</td>
<td>Clare</td>
</tr>
<tr>
<td>Xavier</td>
<td>Bertha</td>
<td>Erika</td>
<td>Clare</td>
<td>Diane</td>
</tr>
<tr>
<td>Yancey</td>
<td>Amy</td>
<td>Diane</td>
<td>Clare</td>
<td>Bertha</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha</td>
<td>Diane</td>
<td>Amy</td>
<td>Erika</td>
</tr>
</tbody>
</table>

Wyatt proposes to Diane.

### Women's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Zeus</td>
<td>Victor</td>
<td>Wyatt</td>
<td>Yancey</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Wyatt</td>
<td>Yancey</td>
<td>Victor</td>
</tr>
<tr>
<td>Clare</td>
<td>Wyatt</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Diane</td>
<td>Victor</td>
<td>Zeus</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Erika</td>
<td>Yancey</td>
<td>Wyatt</td>
<td>Zeus</td>
<td>Xavier</td>
</tr>
</tbody>
</table>

Wyatt proposes to Diane.
- Diane accepts since previously unmatched.
### Men's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victor</td>
<td>Bertha</td>
<td>Amy</td>
<td>Diane</td>
<td>Erika</td>
</tr>
<tr>
<td>Wyatt</td>
<td>Diane</td>
<td>Bertha</td>
<td>Amy</td>
<td>Clare</td>
</tr>
<tr>
<td>Xavier</td>
<td>Bertha</td>
<td>Erika</td>
<td>Clare</td>
<td>Diane</td>
</tr>
<tr>
<td>Yancey</td>
<td>Amy</td>
<td>Diane</td>
<td>Clare</td>
<td>Bertha</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha</td>
<td>Diane</td>
<td>Amy</td>
<td>Erika</td>
</tr>
</tbody>
</table>

### Women's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Zeus</td>
<td>Victor</td>
<td>Wyatt</td>
<td>Yancey</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Wyatt</td>
<td>Yancey</td>
<td>Victor</td>
</tr>
<tr>
<td>Clare</td>
<td>Wyatt</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Diane</td>
<td>Victor</td>
<td>Zeus</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Erika</td>
<td>Yancey</td>
<td>Wyatt</td>
<td>Zeus</td>
<td>Xavier</td>
</tr>
</tbody>
</table>

Xavier proposes to Bertha.

Bertha dumps Victor and accepts Xavier.

Xavier proposes to Bertha.

- Bertha dumps Victor and accepts Xavier.
<table>
<thead>
<tr>
<th>Men's Preference Profile</th>
<th>Women's Preference Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Victor</strong> AMY</td>
<td><strong>Amy</strong> Zeus Victor Wyatt Yancey Xavier</td>
</tr>
<tr>
<td>Wyatt</td>
<td>Diane Bertha Amy Clare Erika</td>
</tr>
<tr>
<td>Xavier</td>
<td>Bertha Erika Clare Diane Amy</td>
</tr>
<tr>
<td>Yancey</td>
<td>Amy Diane Clare Bertha Erika</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha Diane Amy Erika Clare</td>
</tr>
</tbody>
</table>

Victor proposes to Amy.

Victor proposes to Amy.

- Amy accepts since previously unmatched.

| **Victor** AMY | **Amy** Zeus Victor Wyatt Yancey Xavier |
| Wyatt | Diane Bertha Amy Clare Erika |
| Xavier | Bertha Erika Clare Diane Amy |
| Yancey | Amy Diane Clare Bertha Erika |
| Zeus | Bertha Diane Amy Erika Clare |

Victor proposes to Amy.

Victor proposes to Amy.

- Amy accepts since previously unmatched.
**Men's Preference Profile**

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victor</td>
<td>Amy</td>
<td>Diane</td>
<td>Erika</td>
<td>Clare</td>
</tr>
<tr>
<td>Wyatt</td>
<td>Diane</td>
<td>Bertha</td>
<td>Amy</td>
<td>Clare</td>
</tr>
<tr>
<td>Xavier</td>
<td>Bertha</td>
<td>Amy</td>
<td>Clare</td>
<td>Diane</td>
</tr>
<tr>
<td>Yancey</td>
<td>Amy</td>
<td>Diane</td>
<td>Clare</td>
<td>Bertha</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha</td>
<td>Diane</td>
<td>Amy</td>
<td>Erika</td>
</tr>
</tbody>
</table>

**Women's Preference Profile**

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Zeus</td>
<td>Victor</td>
<td>Wyatt</td>
<td>Yancey</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Wyatt</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Clare</td>
<td>Wyatt</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Diane</td>
<td>Victor</td>
<td>Zeus</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Erika</td>
<td>Yancey</td>
<td>Wyatt</td>
<td>Zeus</td>
<td>Xavier</td>
</tr>
</tbody>
</table>

Yancey proposes to Amy.

- Amy rejects since she prefers Victor.
### Men's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victor</td>
<td>Amy</td>
<td>Diane</td>
<td>Erika</td>
<td>Clare</td>
</tr>
<tr>
<td>Wyatt</td>
<td>Diane</td>
<td>Bertha</td>
<td>Amy</td>
<td>Clare</td>
</tr>
<tr>
<td>Xavier</td>
<td>Bertha</td>
<td>Erika</td>
<td>Clare</td>
<td>Diane</td>
</tr>
<tr>
<td>Yancey</td>
<td><strong>Diane</strong></td>
<td>Clare</td>
<td>Bertha</td>
<td>Erika</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha</td>
<td>Diane</td>
<td>Amy</td>
<td>Erika</td>
</tr>
</tbody>
</table>

### Women's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Zeus</td>
<td><strong>Victor</strong></td>
<td>Wyatt</td>
<td>Xavier</td>
</tr>
<tr>
<td>Bertha</td>
<td><strong>Xavier</strong></td>
<td>Wyatt</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Clare</td>
<td>Wyatt</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Diane</td>
<td>Victor</td>
<td>Zeus</td>
<td><strong>Yancey</strong></td>
<td>Xavier</td>
</tr>
<tr>
<td>Erika</td>
<td>Yancey</td>
<td>Wyatt</td>
<td>Zeus</td>
<td>Xavier</td>
</tr>
</tbody>
</table>

- Yancey proposes to Diane.

- Diane dumps Wyatt and accepts Yancey.
Wyatt proposes to Bertha.

- Bertha rejects since she prefers Xavier.
Wyatt proposes to Amy.

- Amy rejects since she prefers Victor.
Men's Preference Profile

<table>
<thead>
<tr>
<th>0th</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victor</td>
<td>Amy</td>
<td>Diane</td>
<td>Erika</td>
<td>Clare</td>
</tr>
<tr>
<td>Wyatt</td>
<td>Bertha</td>
<td>Erika</td>
<td>Clare</td>
<td>Diane</td>
</tr>
<tr>
<td>Xavier</td>
<td>Yancey</td>
<td>Diane</td>
<td>Clare</td>
<td>Bertha</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha</td>
<td>Diane</td>
<td>Amy</td>
<td>Erika</td>
</tr>
</tbody>
</table>

Women's Preference Profile

<table>
<thead>
<tr>
<th>0th</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Zeus</td>
<td>Victor</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
<td>Victor</td>
</tr>
<tr>
<td>Clare</td>
<td>Wyatt</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Diane</td>
<td>Victor</td>
<td>Zeus</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Erika</td>
<td>Yancey</td>
<td>Wyatt</td>
<td>Zeus</td>
<td>Xavier</td>
</tr>
</tbody>
</table>

Wyatt proposes to Clare.

Women's Preference Profile

<table>
<thead>
<tr>
<th>0th</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Zeus</td>
<td>Victor</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
<td>Victor</td>
</tr>
<tr>
<td>Clare</td>
<td>Wyatt</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Diane</td>
<td>Victor</td>
<td>Zeus</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Erika</td>
<td>Yancey</td>
<td>Wyatt</td>
<td>Zeus</td>
<td>Xavier</td>
</tr>
</tbody>
</table>

Wyatt proposes to Clare.

- Clare accepts since previously unmatched.
Zeus proposes to Bertha.

Zeus proposes to Bertha.

Zeus proposes to Bertha.

Zeus proposes to Bertha.

- Bertha rejects since she prefers Xavier.

---

**Men’s Preference Profile**

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victor</td>
<td>Amy</td>
<td>Diane</td>
<td>Erika</td>
<td>Clare</td>
</tr>
<tr>
<td>Wyatt</td>
<td>Bertha</td>
<td>Erika</td>
<td>Clare</td>
<td>Diane</td>
</tr>
<tr>
<td>Xavier</td>
<td>Clare</td>
<td>Diane</td>
<td>Bertha</td>
<td>Erika</td>
</tr>
<tr>
<td>Yancey</td>
<td>Diane</td>
<td>Amy</td>
<td>Erika</td>
<td>Clare</td>
</tr>
<tr>
<td>Zeus</td>
<td>Bertha</td>
<td>Xavier</td>
<td>Zeusa</td>
<td>Wyatt</td>
</tr>
</tbody>
</table>

**Women’s Preference Profile**

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Zeus</td>
<td>Victor</td>
<td>Erika</td>
<td>Clare</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
<td>Victor</td>
</tr>
<tr>
<td>Clare</td>
<td>Wyatt</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeusa</td>
</tr>
<tr>
<td>Diane</td>
<td>Victor</td>
<td>Zeus</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Erika</td>
<td>Yancey</td>
<td>Wyatt</td>
<td>Zeusa</td>
<td>Xavier</td>
</tr>
</tbody>
</table>

- Bertha rejects since she prefers Xavier.
Zeus proposes to Diane.

Zeus proposes to Diane.

- Diane rejects Yancey and accepts Zeus.
Yancey proposes to Clare.
- Clare rejects since she prefers Wyatt.
Yancey proposes to Bertha.

- Bertha rejects since she prefers Xavier.
Yancey proposes to Erika.

Yancey proposes to Erika.
- Erika accepts since previously unmatched.
We want to prove the algorithm always finds a stable matching.

**Definition:** a proof is a convincing demonstration (within the accepted standards of the field) that some mathematical statement is necessarily true.

"A proof is a proof. What kind of a proof? It's a proof. A proof is a proof. And when you have a good proof, it's because it's proven."

- Jean Chretien

Need to show:
1. The algorithm terminates.
2. We get a perfect matching.
3. The matching is stable.
Proof of Correctness: Termination

Observation 1. Men propose to women in decreasing order of preference.

Observation 2. Once a woman is matched, she never becomes unmatched; she only “trades up.”

Claim. Algorithm terminates after at most $n^2$ iterations of while loop.

Pf. (direct proof)

By the observations above, each time through the while loop a man proposes to a new woman. There are only $n^2$ possible proposals. Thus, the algorithm must terminate.

Proof of Correctness: Perfection

Claim. All men and women get matched.

Pf. (by contradiction)

Suppose, for sake of contradiction, that Zeus is not matched upon termination of algorithm.

Then some woman, say Amy, is not matched upon termination. By Observation 2, Amy was never proposed to.

But, Zeus proposes to everyone, since he ends up unmatched.

Hence, a contradiction!
Proof of Correctness: Stability

Claim. No unstable pairs.

Pf. (direct proof)

Suppose A-Z are not engaged to each other.

Case 1: Z never proposed to A.
⇒ Z prefers his GS partner to A.
⇒ A-Z is stable.

Case 2: Z proposed to A.
⇒ A rejected Z (right away or later)
⇒ A prefers her GS partner to Z.
⇒ A-Z is stable.

In either case A-Z is stable.
Understanding the Solution

Q. For a given problem instance, there may be several stable matchings. Do all executions of Gale-Shapley yield the same stable matching? If so, which one?

Def. Man m is a valid partner of woman w if there exists some stable matching in which they are matched.

Man-optimal assignment. Each man receives best valid partner.

Claim. All executions of GS yield man-optimal assignment, which is a stable matching.

- No reason a priori to believe that man-optimal assignment is perfect, let alone stable!
- Simultaneously best for each and every man!
- Order of proposals does not affect outcome!

Man Optimality

Claim. GS matching \( S^* \) is man-optimal.

Pf. (by contradiction)

- Suppose some man is paired with someone other than best partner.
- Men propose in decreasing order of preference, so some man is rejected by valid partner.
  - Let \( Y \) be the first such man, and let \( A \) be the first valid woman that rejects him.
  - Let \( S \) be any matching where \( A \) and \( Y \) are matched.
  - When \( Y \) is rejected, \( A \) reaffirms engagement with a man, say \( Z \), whom she prefers to \( Y \).
    - Let \( B \) be \( Z \)'s partner in \( S \).
    - \( Z \) is not rejected by any valid partner at the point when \( Y \) is rejected by \( A \).
      - Thus, \( Z \) prefers \( A \) to \( B \).
      - But \( A \) prefers \( Z \) to \( Y \).
        - Thus \( A-Z \) is unstable.
          - This contradicts the validity of \( A \).
Woman Pessimality

**Woman-pessimal assignment.** Each woman receives worst valid partner.

**Claim.** GS finds woman-pessimal stable matching $S^*$.

**Pf.** *(direct proof)*

1. Suppose $A-Z$ is matched in $S^*$.
   - Let $S$ be any matching in which $A$ is paired with a man, say $Y$, whom she likes less than $Z$.
   - Let $B$ be $Z$’s partner in $S$.
   - $Z$ prefers $A$ to $B$.  --- man-optimality
   - Thus, $A-Z$ is unstable in $S$.
   - Thus $Y$ is not a valid partner for $A$.
   - Hence, $Z$ is $A$'s worst valid partner.

---

Summary

**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 problem instance.

Q. If there are multiple stable matchings, which one does GS find?

Q. How to implement GS algorithm efficiently?

Will come back to this idea.
Lessons Learned

Powerful ideas learned in course.
- Isolate underlying structure of problem.
- Create useful and efficient algorithms.

Potentially deep social ramifications. [legal disclaimer]
- Historically, men propose to women. Why not vice versa?
- Men: propose early and often.
- Men: be more honest.
- Women: ask out the guys.
- Theory can be socially enriching!

Extensions: Matching Residents to Hospitals

Ex: Men = hospitals, Women = med school residents.

Variant 1. Some participants declare others as unacceptable.

Variant 2. Unequal number of men and women.

Variant 3. Limited polygamy.

Def. Matching S unstable if there is a hospital h and resident r such that:
- h and r are acceptable to each other; and
- either r is unmatched, or r prefers h to her assigned hospital; and
- either h does not have all its places filled, or h prefers r to at least one of its assigned residents.
Extensions: Stable Roommate Problem

Q. Do stable matchings always exist?
A. Not obvious a priori.

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

**Observation.** Stable matchings do not always exist for stable roommate problem.

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>Bob</td>
<td>C</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>Chris</td>
<td>A</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>Doofus</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

A-B, C-D $\Rightarrow$ B-C unstable
A-C, B-D $\Rightarrow$ A-B unstable
A-D, B-C $\Rightarrow$ A-C unstable

Extensions: Deceit! (Machiavelli Meets Gale-Shapley)

Q. Can there be an incentive to misrepresent your preference profile?
- Assume you know men's propose-and-reject algorithm will be run.
- Assume that you know the preference profiles of all other participants.

A. Problem 1.8 (Extra Credit).
2.1 Computational Tractability

“As soon as an Analytic Engine exists, [...] whenever any result is sought by its aid, the question will arise: By what course of calculation can these results be arrived at by the machine in the shortest time?”

- Charles Babbage
Worst-Case Analysis

Worst case running time. Obtain bound on largest possible running time of algorithm on input of a given size $N$.
- Generally captures efficiency in practice.
- Draconian view, but hard to find effective alternative.

Average case running time. Obtain bound on running time of algorithm on random input as a function of input size $N$.
- Hard (or impossible) to accurately model real instances by random distributions.
- Algorithm tuned for a certain distribution may perform poorly on other inputs.

Polynomial-Time

Brute force. For many non-trivial problems, there is a natural brute force search algorithm that checks every possible solution.
- Typically takes $2^N$ time or worse for inputs of size $N$.
- Unacceptable in practice.

Desirable scaling property. When the input size doubles, the algorithm should only slow down by some constant factor $C$.

There exists constants $c > 0$ and $d > 0$ such that on every input of size $N$, its running time is bounded by $c N^d$ steps.

Def. An algorithm is poly-time if the above scaling property holds.
Worst-Case Polynomial-Time

**Def.** An algorithm is **efficient** if its running time is polynomial.

**Justification:** It really works in practice!
- Although $6.02 \times 10^{23} \times N^{20}$ is technically poly-time, it would be useless in practice.
- In practice, the poly-time algorithms that people develop almost always have low constants and low exponents.
- Breaking through the exponential barrier of brute force typically exposes some crucial structure of the problem.

**Exceptions.**
- Some poly-time algorithms do have high constants and/or exponents, and are useless in practice.
- Some exponential-time (or worse) algorithms are widely used because the worst-case instances seem to be rare.

---

### Why It Matters

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n \log n$</th>
<th>$n^2$</th>
<th>$n^3$</th>
<th>$1.3^n$</th>
<th>$2^n$</th>
<th>$n!$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>4 sec</td>
</tr>
<tr>
<td>30</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>18 min</td>
<td>$10^{23}$ years</td>
</tr>
<tr>
<td>50</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>11 min</td>
<td>36 years</td>
<td>very long</td>
</tr>
<tr>
<td>100</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>1 sec</td>
<td>12,892 years</td>
<td>$10^{17}$ years</td>
</tr>
<tr>
<td>1,000</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>18 min</td>
<td>very long</td>
<td>very long</td>
<td>very long</td>
</tr>
<tr>
<td>10,000</td>
<td>&lt; 1 sec</td>
<td>2 min</td>
<td>12 days</td>
<td>very long</td>
<td>very long</td>
<td>very long</td>
</tr>
<tr>
<td>100,000</td>
<td>&lt; 1 sec</td>
<td>3 hours</td>
<td>32 years</td>
<td>very long</td>
<td>very long</td>
<td>very long</td>
</tr>
<tr>
<td>1,000,000</td>
<td>1 sec</td>
<td>20 sec</td>
<td>31,700 years</td>
<td>very long</td>
<td>very long</td>
<td>very long</td>
</tr>
</tbody>
</table>
2.2 Asymptotic Order of Growth

**Upper bounds.** \( T(n) \in O(f(n)) \)
If there exist constants \( c > 0 \) and \( n_0 \geq 0 \) such that for all \( n \geq n_0 \) we have
\[
T(n) \leq c \cdot f(n).
\]

**Lower bounds.** \( T(n) \in \Omega(f(n)) \)
If there exist constants \( c > 0 \) and \( n_0 \geq 0 \) such that for all \( n \geq n_0 \) we have
\[
T(n) \geq c \cdot f(n).
\]

**Tight bounds.** \( T(n) \in \Theta(f(n)) \)
If \( T(n) \in O(f(n)) \) and \( T(n) \in \Omega(f(n)) \).

**Ex:** \( T(n) = 32n^2 + 17n + 32 \)
- \( T(n) \in O(n^2), O(n^3), \Omega(n^2), \Omega(n), \) and \( \Theta(n^2) \).
- \( T(n) \in O(n), \Omega(n^2), \Theta(n), \) or \( \Theta(n^3) \).
Important!

*Meaningless statement.* Any comparison-based sorting algorithm requires at least $O(n \log n)$ comparisons.
- Statement doesn’t "type-check."
- Use $\Omega$ for lower bounds.

**Properties**

**Transitivity.**
- If $f \in O(g)$ and $g \in O(h)$ then $f \in O(h)$.
- If $f \in \Omega(g)$ and $g \in \Omega(h)$ then $f \in \Omega(h)$.
- If $f \in \Theta(g)$ and $g \in \Theta(h)$ then $f \in \Theta(h)$.

**Additivity.**
- If $f \in O(h)$ and $g \in O(h)$ then $f + g \in O(h)$.
- If $f \in \Omega(h)$ and $g \in \Omega(h)$ then $f + g \in \Omega(h)$.
- If $f \in \Theta(h)$ and $g \in O(h)$ then $f + g \in \Theta(h)$. 
Asymptotic Bounds for Some Common Functions

Polynomials. \((a_0 + a_1 n + \ldots + a_n n^d) \in \Theta(n^d)\) if \(a_d > 0\).

Polynomial time. Running time is \(O(n^d)\) for some constant \(d\) independent of the input size \(n\).

Logarithms. \(O(\log_a n) = O(\log_b n)\) for any constants \(a, b > 0\).

Logarithms. For every \(x > 0\), \(\log n \in O(n^x)\).

Exponentials. For every \(r > 1\) and every \(d > 0\), \(n^d \in O(n^r)\).
2.4 A Survey of Common Running Times

Linear Time: $O(n)$

Linear time. Running time is proportional to input size.

Computing the maximum. Compute maximum of $n$ numbers $a_1, \ldots, a_n$.

```plaintext
max ← $a_1$
for $i = 2$ to $n$ {
    if ($a_i > max$)
        max ← $a_i$
}
```
**Linear Time: \( O(n) \)**

**Merge.** Combine two sorted lists \( A = a_1, a_2, \ldots, a_n \) with \( B = b_1, b_2, \ldots, b_n \) into sorted whole.

\[
i = 1, \ j = 1
\]

\[
\text{while (both lists are nonempty)}(
\text{if } (a_i \leq b_j)
\text{ append } a_i \text{ to output list}
\text{ increment } i
\text{ else (}a_i \leq b_j\text{)}
\text{ append } b_j \text{ to output list}
\text{ increment } j
\}
\]

append remainder of nonempty list to output list

**Claim.** Merging two lists of size \( n \) takes \( O(n) \) time.

**Pf.** After each comparison, the length of output list increases by 1.

---

**\( O(n \log n) \) Time**

**\( O(n \log n) \) time.** Arises in divide-and-conquer algorithms.

**Sorting.** Mergesort and heapsort are sorting algorithms that perform \( O(n \log n) \) comparisons.

**Largest empty interval.** Given \( n \) time-stamps \( x_1, \ldots, x_n \) on which copies of a file arrive at a server, what is largest interval of time when no copies of the file arrive?

**\( O(n \log n) \) solution.** Sort the time-stamps. Scan the sorted list in order, identifying the maximum gap between successive time-stamps.
Quadratic Time: $O(n^2)$

**Quadratic time.** Enumerate all pairs of elements.

**Closest pair of points.** Given a list of $n$ points in the plane $(x_1, y_1), \ldots, (x_n, y_n)$, find the pair that is closest.

$O(n^2)$ solution. Try all pairs of points.

```
min ← (x_1 - x_2)^2 + (y_1 - y_2)^2
for i = 1 to n {
    for j = i+1 to n {
        d ← (x_i - x_j)^2 + (y_i - y_j)^2
        if (d < min) min ← d
    }
}
```

**Remark.** $\Omega(n^2)$ seems inevitable, but this is just an illusion. — see chapter 5

Cubic Time: $O(n^3)$

**Cubic time.** Enumerate all triples of elements.

**Set disjointness.** Given $n$ sets $S_1, \ldots, S_n$ each of which is a subset of 1, 2, ..., $n$, is there some pair of these which are disjoint?

$O(n^3)$ solution. For each pairs of sets, determine if they are disjoint.

```
foreach set S_i {
    foreach other set S_j {
        foreach element p of S_i {
            determine whether p belongs to S_j
        }
        if (no element of S_i belongs to S_j) report that S_i and S_j are disjoint
    }
}
```
Polynomial Time: $O(n^k)$ Time

**Independent set of size $k$.** Given a graph, are there $k$ nodes such that no two are joined by an edge?

$k$ is a constant

$O(n^k)$ solution. Enumerate all subsets of $k$ nodes.

```
foreach subset $S$ of $k$ nodes {
    check whether $S$ is an independent set
    if ($S$ is an independent set)
        report $S$ is an independent set
}
```

- Check whether $S$ is an independent set = $O(k^2)$.
- Number of $k$ element subsets = $\binom{n}{k} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k(k-1)(k-2)\cdots(2)(1)} \leq \frac{n^k}{k!}$
- $O(k^2 \cdot n!) = O(n^k)$, poly-time for $k=17$, but not practical

Exponential Time

**Independent set.** Given a graph, what is maximum size of an independent set?

$O(n^2 2^n)$ solution. Enumerate all subsets.

```
S* ← φ
foreach subset $S$ of nodes {
    check whether $S$ is an independent set
    if ($S$ is largest independent set so far)
        update $S^* ← S$
}
```