Algorithm Design and Analysis

Lecture 2
- Analysis of Stable Matching
- Asymptotic Notation

Adam Smith

A. Smith; based on slides by E. Demaine, C. Leiserson, S. Raskhodnikova, K. Wayne
**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.

### Men's Preference Profile

<table>
<thead>
<tr>
<th>Favorite</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xavier</td>
<td>Amy</td>
<td>Bertha</td>
</tr>
<tr>
<td>Yancy</td>
<td>Bertha</td>
<td>Amy</td>
</tr>
<tr>
<td>Zeus</td>
<td>Amy</td>
<td>Bertha</td>
</tr>
</tbody>
</table>

### Women's Preference Profile

<table>
<thead>
<tr>
<th>Favorite</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Yancy</td>
<td>Xavier</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Yancy</td>
</tr>
<tr>
<td>Clare</td>
<td>Xavier</td>
<td>Yancy</td>
</tr>
</tbody>
</table>

8/27/2008

*A. Smith; based on slides by E. Demaine, C. Leiserson, S. Raskhodnikova, K. Wayne*
Stable Matching Problem

• **Unstable pair**: man $m$ and woman $w$ are **unstable** if
  – $m$ prefers $w$ to his assigned match, and
  – $w$ prefers $m$ to her assigned match

• **Stable assignment**: no unstable pairs.

**Men's Preference Profile**

<table>
<thead>
<tr>
<th>favorite</th>
<th>least favorite</th>
</tr>
</thead>
<tbody>
<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>

**Women's Preference Profile**

<table>
<thead>
<tr>
<th>favorite</th>
<th>least favorite</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>2nd</td>
</tr>
<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>
Review Questions

• In terms of \( n \), what is the length of the input to the Stable Matching problem, i.e., the number of entries in the tables?

  (Answer: \( 2n^2 \) list entries, or \( 2n^2 \log n \) bits)
Review Question

- **Brute force algorithm:** an algorithm that checks every possible solution.

- In terms of \( n \), what is the running time for the brute force algorithm for Stable Matching Problem? (Assume your algorithm goes over all possible perfect matchings.)

(Answer: \( n! \times (\text{time to check if a matching is stable}) = \Theta(n! \ n^2) \))
Propose-and-Reject Algorithm

- Propose-and-reject algorithm. [Gale-Shapley 1962]

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
}
Proof of Correctness: Termination

• **Claim.** Algorithm terminates after at most \( n^2 \) iterations of while loop.

• **Pf.** Each time through the loop a man proposes to a new woman. There are only \( n^2 \) possible proposals.

An instance where \( n(n-1) + 1 \) proposals required

<table>
<thead>
<tr>
<th>Victor</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wyatt</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>A</td>
<td>E</td>
</tr>
<tr>
<td>Xavier</td>
<td>C</td>
<td>D</td>
<td>A</td>
<td>B</td>
<td>E</td>
</tr>
<tr>
<td>Yancey</td>
<td>D</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>E</td>
</tr>
<tr>
<td>Zeus</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Amy</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bertha</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>V</td>
<td>W</td>
</tr>
<tr>
<td>Clare</td>
<td>Y</td>
<td>Z</td>
<td>V</td>
<td>W</td>
<td>X</td>
</tr>
<tr>
<td>Diane</td>
<td>Z</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>Erika</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

8/27/2008

*A. Smith; based on slides by E. Demaine, C. Leiserson, S. Raskhodnikova, K. Wayne*
Propose-and-Reject Algorithm

• **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."
Proof of Correctness: Perfection

• **Claim.** All men and women get matched.

• **Proof: (by contradiction)**
  
  – Suppose, for sake of contradiction, some guy, say 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. □
Proof of Correctness: Stability

• **Claim.** No unstable pairs.

• **Proof:** (by contradiction)
  
  – Suppose A-Z is an unstable pair: they prefer each other to their partners in Gale-Shapley matching S*.

  – **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, a contradiction. □
Efficient Implementation

- We describe $O(n^2)$ time implementation.
- Assume men have IDs 1,…, $n$, and so do women.
- Engagements data structures:
  - a list of free men, e.g., a queue.
  - two arrays $\text{wife}[m]$, and $\text{husband}[w]$.
    - set entry to 0 if unmatched
    - if $m$ matched to $w$ then $\text{wife}[m] = w$ and $\text{husband}[w] = m$
- Men proposing data structures:
  - an array $\text{men-pref}[m, i] =$ $i^{\text{th}}$ women on $m^{\text{th}}$ list
  - an array $\text{count}[m] =$ how many proposals $m$ made.
Efficient Implementation

- Women rejecting/accepting data structures
  - Does woman $w$ prefer man $m$ to man $m'$?
  - For each woman, create inverse of preference list of men.
  - Constant time queries after $O(n)$ preprocessing per woman.

```
for i = 1 to n
    inverse[pref[i]] = i
```

<table>
<thead>
<tr>
<th>Amy</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>6th</th>
<th>7th</th>
<th>8th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pref</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Amy</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverse</td>
<td>4th</td>
<td>8th</td>
<td>2nd</td>
<td>5th</td>
<td>6th</td>
<td>7th</td>
<td>3rd</td>
<td>1st</td>
</tr>
</tbody>
</table>

Amy prefers man 3 to 6 since $\text{inverse}[3] < \text{inverse}[6]$

A. Smith; based on slides by E. Demaine, C. Leiserson, S. Raskhodnikova, K. Wayne
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 every problem instance.
  – (Also proves that stable matching always exists)

• **Time and space complexity:** \( O(n^2) \), linear in the input size.
Brief Syllabus

• Reminders
  – Worst-case analysis
  – Asymptotic notation
  – Basic Data Structures

• Design Paradigms
  – Greedy algorithms, Divide and conquer, Dynamic programming, Network flow and linear programming

• Analyzing algorithms in other models
  – Parallel algorithms, Memory hierarchies (?)

• P, NP and NP-completeness

8/27/2008

A. Smith; based on slides by E. Demaine, C. Leiserson, S. Raskhodnikova, K. Wayne
Measuring Running Time

• Focus on **scalability**: parameterize the running time by some measure of “size”
  – (e.g. \( n \) = number of men and women)

• Kinds of analysis
  – Worst-case
  – Average-case (requires knowing the distribution)
  – Best-case (how meaningful?)

• Exact times depend on computer; instead measure **asymptotic growth**