CSE 421: Introduction to Algorithms

Yin-Tat Lee
Quiz

Input: two $n$-size strings $s_1$ and $s_2$.
- $s_1 = AGGCTACC$
- $s_2 = CAGGCTAC$

Output: minimum number of insertions/deletions to transform $s_1$ into $s_2$.

Algorithm: ????

Naively, $O(1)^n$ by enumerating all changes.

After this quarter, you should be able to
- solve it in $O(n^2)$ time
- explain why your algorithm is correct.

Question: Is $O(n^{1.999})$ possible?
This Course

Solve computational problems using less steps.

Goal:
• Learn some techniques to design algorithms
• Understand some problems are difficult
• Prove some correctness

Grading Scheme
• Homework 50%
  Weekly homework due on Wed before the class
• Midterm 20%
• Final 30%

You will get at least 3.4 if
• Your score is >= 65%

Do not come to the class/exam if you feel sick! (There are videos)
Where to get help?

• Ask questions in the class!
• Read the textbook!
• Edstem: Online discussion forum (To be online).
• Office hours:
  • Myself: M 2:30-3:30 in CSE 562

Starting next week

Website: cs.washington.edu/421
Stable Matching Problem

Given \( n \) men and \( n \) women, find a “stable matching”.

- We know the preference of all people.

<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>1\textsuperscript{st}</td>
<td>2\textsuperscript{nd}</td>
</tr>
<tr>
<td>Xavier</td>
<td>Amy</td>
</tr>
<tr>
<td>Yuri</td>
<td>Brenda</td>
</tr>
<tr>
<td>Zoran</td>
<td>Amy</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>favorite</strong></td>
<td><strong>least favorite</strong></td>
</tr>
<tr>
<td>1\textsuperscript{st}</td>
<td>2\textsuperscript{nd}</td>
</tr>
<tr>
<td>Amy</td>
<td>Yuri</td>
</tr>
<tr>
<td>Brenda</td>
<td>Xavier</td>
</tr>
<tr>
<td>Claire</td>
<td>Xavier</td>
</tr>
</tbody>
</table>
Stable Matching

**Perfect matching:**
- Each man gets exactly one woman.
- Each woman gets exactly one man.

**Stability:** no incentive to exchange
- an unmatched pair \( m-w \) is unstable if \( m \) and \( w \) prefer each other to their current partners.

An unstable match:
ELIZABETH AND DARCY LIKE EACH OTHER BETTER THAN THEIR PARTNERS
Stable Matching

Perfect matching:
• Each man gets exactly one woman.
• Each woman gets exactly one man.

Stability: no incentive to exchange
• an unmatched pair $m$-$w$ is unstable
  if $m$ and $w$ prefer each other to their current partners.

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.
Example

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

Men's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xavier</td>
<td>Amy</td>
<td>Brenda</td>
<td>Claire</td>
</tr>
<tr>
<td>Yuri</td>
<td>Brenda</td>
<td>Amy</td>
<td>Claire</td>
</tr>
<tr>
<td>Zoran</td>
<td>Amy</td>
<td>Brenda</td>
<td>Claire</td>
</tr>
</tbody>
</table>

Women's Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Yuri</td>
<td>Xavier</td>
<td>Zoran</td>
</tr>
<tr>
<td>Brenda</td>
<td>Xavier</td>
<td>Yuri</td>
<td>Zoran</td>
</tr>
<tr>
<td>Claire</td>
<td>Xavier</td>
<td>Yuri</td>
<td>Zoran</td>
</tr>
</tbody>
</table>
Example

Question. Is assignment X-C, Y-B, Z-A stable?
Answer. No. Look at Brenda and Xavier.
Question: Is assignment X-A, Y-B, Z-C stable?
Answer: Yes. (X, Y are happy. No one want Z.)
Existence of Stable Matchings

Question. Do stable matchings always exist?

Answer. Seems unclear in real-world. Yet, it always exists!

Question. How to find stable matchings?
Propose-And-Reject Algorithm [Gale-Shapley’62]

Initialize each person to be free.

while (some man is free) {
    Choose such a man $m$
    $w = 1^{st}$ 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 $m$
}

Switch the pdf for an example.
Basic Properties

- The algorithm ends.
  Is every step valid?
  How many iterations it takes?

- The output is correct.
  It finds a **perfect** matching that is **stable**.

**How to prove them?**
When stuck, try to list out properties of the algorithms.

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

**Observation 2:** Woman's partner get better and better. Never unmatched once matched.

```java
Initialize each person to be free.
while (some man is free) {
  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
}
```
Summary

• **Stable matching problem:** Given $n$ men and $n$ women, and their preferences, find a stable matching.

• **Propose-And-Reject algorithm:** Guarantees to find a stable matching for any problem instance in $O(n^2)$ steps.

• **Steps on solving problems:**
  • Formulate the problem
  • Define the algorithm
  • Prove the algorithm ends
  • Prove the output is correct
Why this problem is important?

In 1962, Gale and Shapley published the paper “College Admissions and the Stability of Marriage” To “The American Mathematical Monthly”
Why this problem is important?

Alvin Roth modified the Gale-Shapley algorithm and apply it to
• National Residency Match Program (NRMP), a system that assigns new doctors to hospitals around the country. (90s)

• Public high school assignment process (00s)

• Helping transplant patients find a match (2004) (Saved >1,000 people every year!)

Reference: https://medium.com/@UofCalifornia/how-a-matchmaking-algorithm-saved-lives-2a65ac448698
Why this problem is important?

Some of the problems in this course may seem obscure.
But their abstraction allows for variety of applications.

Shapley and Roth got the Nobel Prize (Economic) in 2012.
(David Gale passed away in 2008.)