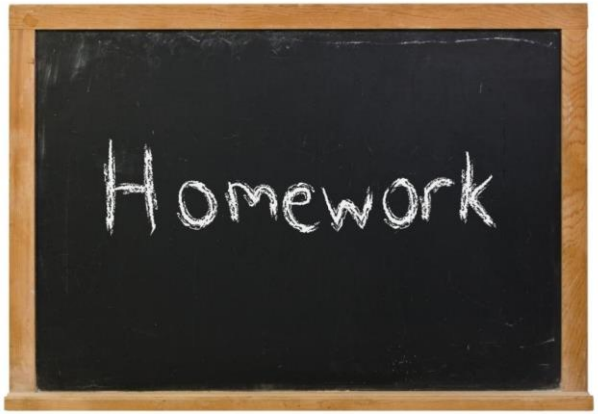
CSE 421: Introduction to Algorithms

Stable Matching

Yin-Tat Lee

Administrativia Stuffs



HW1 is out!
It is due Wednesday Apr 04 before class.

Please submit to Canvas

How to submit?

- Submit a separate file for each problem
- Double check your submission before the deadline!!
- For hand written solutions, take a picture, turn it into pdf and submit

Guidelines:

- Always prove your algorithm halts and outputs correct answer
- You can collaborate, but you must write solutions on your own
- Your proofs should be clear, well-organized, and concise. Spell out main idea.
- Sanity Check: Make sure you use assumptions of the problem
- You CANNOT search the solution online.

Last Lecture (summary)

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

For a perfect matching **M**, a pair **m-w** is unstable if they prefer each other to their match in **M.**

Gale-Shapley algorithm: Guarantees always finds a stable matching by running at most n^2 proposals.

Main properties:

- Men go down their lists
- Women trade up!

Questions

- Q: How to implement GS algorithm efficiently?
- Q: If there are multiple stable matchings, which one does GS find?
- Q: How many stable matchings are there?

Implementation of GS Algorithm

Problem size

$N=2n^2$ words

- **$2n$** people each with a preference list of length **n**

$2n^2 \log n$ bits

- specifying an ordering for each preference list takes **$n \log n$** bits

Q. Why do we care?
A. Usually, the running time is lower-bounded by input length.

Propose-And-Reject Algorithm [Gale-Shapley'62]

```
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
}
```

Efficient Implementation

We describe $O(n^2)$ time implementation.

Representing men and women:

Assume men are named **1**, …, **n**.

Assume women are named **n+1**, …, 2**n**.

Engagements.

Maintain a list of free men, e.g., in a queue.

Maintain two arrays **wife**[**m**], and **husband**[**w**].

- set entry to **0** if unmatched
- if **m** matched to **w** then **wife**[**m**]=**w** and **husband**[**w**]=**m**

Men proposing:

For each man, maintain a list of women, ordered by preference.

Maintain an array **count**[**m**] that counts the number of proposals made by man **m**.

A Preprocessing Idea

Women rejecting/accepting.

Does woman **w** prefer man **m** to man **m'**?

For each woman, create inverse of preference list of men.

Constant time access for each query after **O(n)** preprocessing per woman.

O(n^2) total reprocessing cost.

Amy	1st	2nd	3rd	4th	5th	6th	7th	8th
Pref	8	3	7	1	4	5	6	2

Amy	1	2	3	4	5	6	7	8
Inverse	4th	8th	2nd	5th	6th	7th	3rd	1st

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

Amy prefers man **3** to **6**
since $\mathbf{inverse[3] = 2 < 7 = inverse[6]}$

Questions

- How to implement GS algorithm efficiently?

  We can implement GS algorithm in **$O(n^2)$** time.

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

- Q: How many stable matchings are there?

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?

An instance with two stable matchings:

- A-X, B-Y.
- A-Y, B-X.

	1st	2nd
Xavier	A	B
Yuri	B	A

	1st	2nd
Amy	Y	X
Brenda	X	Y

Man Optimal Assignments

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

Man-optimal matching: Each man receives the best valid partner (according to his preferences).

- Simultaneously best for each and every man.

Claim: All executions of GS yield a man-optimal matching, which is a stable matching!

No reason a priori to believe that man-optimal matching is perfect, let alone stable.

Man Optimality

S
m-w
m'-w'
. . .

Claim: GS matching **S*** is man-optimal.

Proof: (by contradiction)

Suppose some man is paired with someone other than his best partner. Men propose in decreasing order of preference ⇒ some man is rejected by a valid partner.

Let **m** be the man who is the first such rejection, and let **w** be the women who is first valid partner that rejects him.

Let **S** be a stable matching where **w** and **m** are matched.

In building **S*,** when **m** is rejected, **w** forms (or reaffirms) engagement with a man, say **m'**, whom she prefers to **m**.

Let **w'** be the partner of **m'** in **S**.

In building **S*, m'** is not rejected by any valid partner at the point when **m** is rejected by **w**. Thus, **m'** prefers **w** to **w'**.

since this is the **first** rejection by a valid partner

But **w** prefers **m'** to **m**.

Thus **w-m'** is unstable in **S**. ■

Man Optimality Summary

Man-optimality: In version of GS where men propose, each man receives the best valid partner.

> **w** is a valid partner of **m** if there exist some stable matching where **m** and **w** are paired

Q: Does man-optimality come at the expense of the women?

Woman Pessimality

Woman-pessimal assignment: Each woman receives the worst valid partner.

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

Proof.

- Suppose **m-w** matched in **S***, but **m** is not worst valid partner for **w**.
- There exists stable matching **S** in which **w** is paired with a man, say **m'**, whom she likes less than **m**.
- Let **w'** be the partner of **m'** in **S**.
- **m** prefers **w** to **w'**. ← man-optimality of **S***
- Thus, **m-w** is an unstable in **S**.

■

Questions

- Q: How to implement GS algorithm efficiently?
  We can implement GS algorithm in **$O(n^2)$** time.

- Q: If there are multiple stable matchings, which one does GS find?
  It finds the man-optimal woman-pessimal matching.

- Q: How many stable matchings are there?

How many stable Matchings?

We already show every instance has at least 1 stable matchings.

There are instances with about 2.28^n stable matchings.

There are at most 131072^n stable matchings.

A Simply Exponential Upper Bound on the Maximum Number of Stable Matchings

Anna R. Karlin
University of Washington
karlin@cs.washington.edu

Shayan Oveis Gharan
University of Washington
shayan@cs.washington.edu

Robbie Weber
University of Washington
rtweber2@cs.washington.edu

November 9, 2017

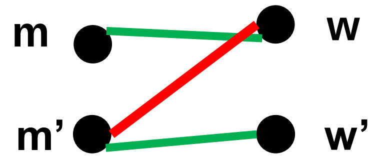
[Research-Question]:

Is there an "efficient" algorithm that chooses a uniformly random stable matching of a given instance.

More realistic

Let $F \subset M \times W$ be the set of relationship (possible marriage).

Can we find a perfect stable matching in F?



No. Consider w prefers m' to m, m' prefers w to w'.
There is no perfect stable matching even there's prefect matching in F.

What can we find?

Gale-Shapley algorithm is very robust to variations!

Same algorithm gives a (non-perfect) matching that is stable.

For a matching **M**, a pair **m-w** is unstable
if **(m,w)**∈ $\boldsymbol{F}$ and they prefer each other to their match in **M.**
(They prefer not to be alone.)