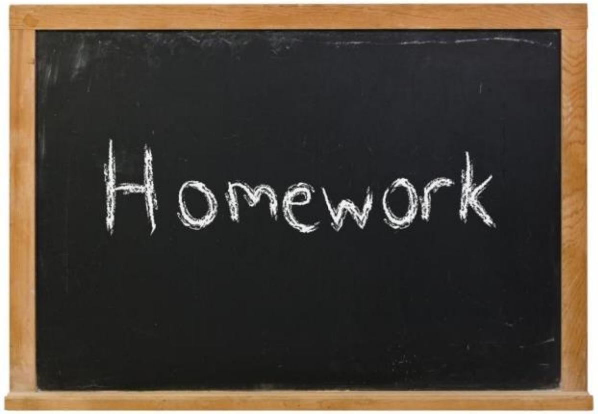
CSE 421: Introduction to Algorithms

Stable Matching/Course Overview

Shayan Oveis Gharan

Administrativia Stuffs



HW1 is out!
It is due Thursday Jan 11 at 5:00

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

Last Lecture (summary)

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

For a perfect matching **M**, a pair (**Z,A)** is unstable
If **Z to A** pair and 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.

Gale-Shapley Algorithm

n^2 proposals, each costing constant time as follows:

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 $\text{inverse}[3] = 2 < 7 = \text{inverse}[6]$

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
Amy-Yuri
Brenda-Zoran
. . .

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 $\Rightarrow$ some man is rejected by a valid partner.

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

Let **S** be a stable matching where **A** and **Y** are matched.

In building **S*,** when **Y** is rejected, **A** forms (or reaffirms) engagement with a man, say **Z**, whom she prefers to **Y**.

Let **B** be **Z**'s partner in **S**.

In building **S*,** **Z** is not rejected by any valid partner at the point when **Y** is rejected by **A**. Thus, **Z** prefers **A** to **B**.

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

But **A** prefers **Z** to **Y**.

Thus **A-Z** 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 **A**-**Z** matched in **S***, but **Z** is not worst valid partner for **A**.
- There exists stable matching **S** 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 of **S***
- Thus, **A**-**Z** 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 c^n stable matchings for $c > 2$

[Research-Question]:

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

Extensions: Matching Residents to Hospitals

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.

e.g. A resident not interested in Cleveland

e.g. A hospital wants to hire **3** residents

Def: Matching **S** is unstable if there is hospital **h** and resident **r** s.t.

- **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.

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.
  - Women: ask out the guys.
  - Theory can be socially enriching and fun!

"The Match": Doctors and Medical Residences

- Each medical school graduate submits a ranked list of hospital where he wants to do a residency

- Each hospital submits a ranked list of newly minted doctors

- A computer runs stable matching algorithm (extended to handle polygamy)

- Until recently, it was hospital-optimal.

History

1900

- Idea of hospital having residents (then called “interns”)

1900-1940s

- Intense competition among hospitals
  - Each hospital makes offers independently
  - Process degenerates into a race; hospitals advancing date at which they finalize binding contracts

1944

- Medical schools stop releasing info about students before a fixed date

1945-1949

- Hospitals started putting time limits on offers
  - Time limits down to 12 hours; lots of unhappy people

“The Match”

1950

- NICI run a centralized algorithm for a trial run
- The pairing was not stable, Oops!!

1952

- The algorithm was modified and adopted. It was called the Match.
- The first matching produced in April 1952