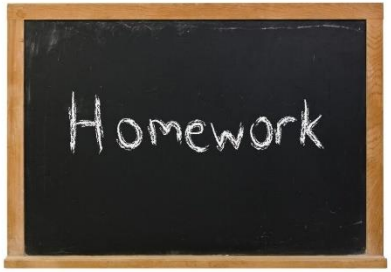
CSE 421: Introduction to Algorithms

Stable Matching

Yin-Tat Lee

Administrativia Stuffs



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:

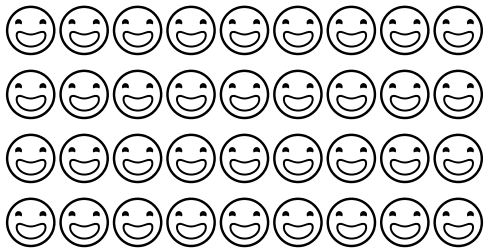
- Always prove your algorithm halts and outputs correct answer
- You can collaborate, but you must write solutions on your own
- You CANNOT search the solution online.

Tips:

- Rewrite your proof. (I often rewrite my proof many times.)
- Explain sol'n to a friend. Let him be an adversary to ask you questions.
- Sanity Check: Make sure you use assumptions of the problem
- Don't write full program. Use pseudo code.

Suggested Guideline

Algorithm:

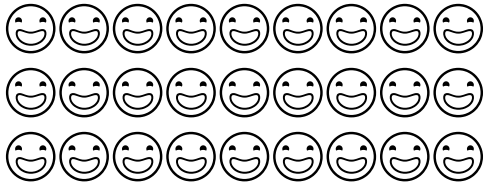


- If you can reduce the question into a solved problem, Just use it. (It prevents mistakes and saves time.)
  For example, for stable matching, explain
  - What “man” and “woman” are corresponding to
  - What their preference are.
  - How convert the stable matching to what we asked in the question.

Runtime:

- If the algorithm is similar to algorithm we teach in class
  You can simply explain what is the difference.
  Make sure your description is not ambiguous.
- Make sure the answer is concise.

Correctness:



- Okay to use a bullet format like my proof.

Last Lecture (summary)

Stable matching problem: Given $\boldsymbol{n}$ men and $\boldsymbol{n}$ women, and their preferences, find a stable matching.

For a perfect matching $\boldsymbol{M}$, a pair $\boldsymbol{m}$-$\boldsymbol{w}$ is unstable if they prefer each other to their match in $\boldsymbol{M}$.

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?

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
}
```

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

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

Have a linked list to store free man

Explained this is not needed

```
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
}
```

$Count[m]$ store how many women m proposed

$W[w,m]$ store the rank of m in the w preference list

$O(1)$ time

$O(n^2)$ iter

Questions

- Q: How to implement GS algorithm efficiently?

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

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. (man optimal)
- A-Y, B-X. (woman optimal)

	1st	2nd
X	A	B
Y	B	A

	1st	2nd
A	Y	X
B	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.

	1st	2nd
X	A	B
Y	B	A

	1st	2nd
A	Y	X
B	X	Y

two stable matchings:

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

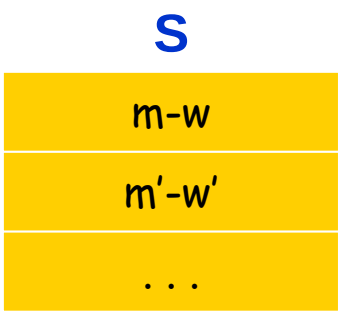
Man-optimal matching: Each man receives the best valid partner (according to his preferences).

- 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



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 rejected by a valid partner, 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.

Fairness Issue

Proposers get a better result.

- Be proactive!

The G-S algorithm looks innocent, turns out to be bloody biased.

To design an algorithm, speed is not everything!

- Efficiency (every one get matched)
- Fairness (do not bias on one-sided)
- Truthfulness (lying on your preference does not help)

Example: http://www.spliddit.org/