CS CS 501
Algorithms: Theory and Practice
Introduction 01.2

Sanjay Rajopadhye
(inspired by Wim Bohm)
Colorado State University
Aug 2023

Questions

- Given lists of preferences or each man and woman does a stable matching even exist?
- If so, is it unique, or how many are there?
- Can we construct it (i.e., an algorithm)?

How about this one:

```
for = S in the set of all perfect matching
  if S is stable return S
Return the empty set
```

- ❑ Is it correct?
- ❑ What is its running time?

Towards an efficient algorithm

- Initially no match
- An unmatched man m *proposes* to the woman w who is the *highest* on his list
- Will this be part of a stable matching?
  - *Not necessarily*,
    - w may like some other m′ better than m
    - and m′ better likes w best
- So this is just one aspect
- *Engagement* – a temporary matching that may be broken
  - w is prepared to change her mind if/when a man *higher on her list* proposes

while anyone is unmatched …

- An unmatched man m proposes to the woman w who is the highest *remaining* on his list (i.e., to whom he hasn't yet proposed)
  - Why is this important?
  - Termination
- If w is free, they become engaged
- If w is engaged to some m′, and
  - m′ is higher than m on w's list – no change
  - Otherwise m and w become engaged and m becomes free

The Gayle-Shapley algorithm[1]

```
Initialize each person to be free.
while (some man is free and hasn't proposed to every woman)
    Choose such a man m
    w = highest-ranked woman on m's list to whom m has not yet proposed
    if (w is free)
        (m,w) become engaged
    else if (w prefers m to her fiancé m')
        (m,w) become engaged, m′ becomes free
    else
        m remains free
```

A few non-obvious questions:

- How long does it take?
- Does the algorithm return a stable matching?
- Does it even return a perfect matching?

[1]D. Gale and L. S. Shapley: "College Admissions and the Stability of Marriage", American Mathematical Monthly 69, 9-14, 1962.

Observations

```
Initialize each person to be free.
while (some man is free and hasn't proposed to every woman)
    Choose such a man m
    w = highest-ranked woman on m's list to whom m has not yet proposed
    if (w is free)
        (m,w) become engaged
    else if (w prefers m to her fiancé m')
        (m,w) become engaged, m' becomes free
    else
        m remains free
```

- Each woman remains engaged from the first proposal she receives and her sequence of partners only improves
- Each man proposes to less and less preferred women
- No man proposes twice to the same woman

Claim 1: complexity

```
Initialize each person to be free.
while (some man is free and hasn't proposed to every woman)
    Choose such a man m
    w = highest-ranked woman on m's list to whom m has not yet proposed
    if (w is free)
        (m,w) become engaged
    else if (w prefers m to her fiancé m')
        (m,w) become engaged, m' becomes free
    else
        m remains free
```

- The algorithm terminates after at most n^2 iterations of the while loop
  - At each iteration, a man proposes (only once) to a woman he has never proposed to
  - each man has only n choices
  - Collectively the n men have n^2 choices

Claim 2: correctness 1

```
Initialize each person to be free.
while (some man is free and hasn't proposed to every woman)
    Choose such a man m
    w = highest-ranked woman on m's list to whom m has not yet proposed
    if (w is free)
        (m,w) become engaged
    else if (w prefers m to her fiancé m')
        (m,w) become engaged, m' becomes free
    else
        m remains free
```

- When the algorithm terminates the matching is perfect (i.e., complete)
  - Proof by contradiction Assume there is a free man m
    - Because the algorithm terminates m must have proposed to all women
    - But then all women are engaged
    - Hence there is no free man   Contradiction

Proof of correctness 2: stability

Claim: When the algorithm terminates, there are no unstable pairs in the Gale-Shapley matching S^*

Proof (by contradiction)

- Suppose some (m, w) is an unstable pair, they each prefer the other to their partner in S^* (see fig)
- Case 1 m never proposed to w

  ⇒ m prefers his GS partner w′ to w

  ⇒ (m, w) is NOT unstable
- Case 2 m proposed to w

  ⇒ w rejected m (right away or later)

  ⇒ w prefers her S^* partner m′ to m

  ⇒ (m, w) is NOT unstable
- In either case (m, w) is NOT unstable
- ⇒ CONTRADICTION

S*
m, w′
m′, w
. . .

Multiple solutions

For an earlier example:

m_1: w_1, w_2 m_2: w_2, w_1

w_1: m_2, m_1 w_2: m_1, m_2

- Two stable solutions

1. { (m_1,w_1), (m_2,w_2) }
2. { (m_1,w_2), (m_2,w_1) }

- GS will always find one of them (which)?
- When will the other be found?

S*
m, w′
m′, w
. . .

Summary

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

- Gale-Shapley algorithm. Guaranteed to find a stable matching for *any* problem instance.

Symmetry

- The stable matching problem is symmetric w.r.t. to men and women, but the GS algorithm is *asymmetric*

- There is a certain unfairness in the algorithm:

  If all men list different women as their first choice, they will end up with their first choice, regardless of the women's preferences (see example 3).

Non determinism

- Notice the line

```
while (some man is free and hasn't proposed to every woman)
    Choose such a man m
...
```

- The algorithm does not specify which

- Nevertheless all executions find the same matching (claim 1.7 in the reading)