Stable Marriages

The problem

Input: n men, n women, each with their own preference list of partners of the opposite sex

Goal: Match up men and women in pairs so that each man and each woman is married to a partner of the opposite sex and all marriages are stable.

Two marriages (m, w) and (m', w') are unstable if m prefers w' over w and w' prefers m over m'.

The proposal algorithm

while there is an unmarried man m
    do m proposes to the next woman w in his preference list
        if w is unmarried or prefers m over her current partner m'
        then w divorces m'
                w marries m

Questions to ask about the (any) algorithm

1. Is there always a solution to the problem we're trying to solve?

2. Does the algorithm always find a solution, i.e., is the algorithm correct?

3. Does the algorithm always terminate?

4. How quickly does the algorithm terminate?

A modified proposal algorithm

```
while there is an unmarried man m who has not
      proposed to all women yet
   do  m proposes to the next woman w in his
       preference list
         if  w is unmarried or prefers m over her
             current partner m'
         then  w divorces m'
               w marries m
```

Lemma: The modified proposal algorithm terminates after at most n^2 iterations.

Proof:
- There are n men.
- Each can propose to n women.
- In each iteration one proposal is made.
- No man proposes to the same woman twice.

□

Lemma: When the modified proposal algorithm terminates, every woman (and hence every man) is married.

Proof: Assume the contrary.
$\Rightarrow$ $\exists$ man m and woman w that are unmarried when the algorithm terminates

Since m is unmarried and the algorithm terminates, m must have exhausted his preference list.
$\Rightarrow$ m must have proposed to w at some point.

w is married after the proposal (not necessarily to m).

A woman, once married, stays married (not necessarily to the same man)
$\Rightarrow$ w must be married when the algorithm terminates, a contradiction. $\square$

Corollary: The standard proposal algorithm terminates after at most n^2 iterations.

Proof: We proved that the modified algorithm terminates after $\leq n^2$ iterations. By the previous lemma, that algorithm never uses its ability to terminate because all preference lists are exhausted. Thus, it behaves the same as the original algorithm.
$\Rightarrow$ The original algorithm terminates after $\leq n^2$ iterations. $\square$

Lemma: The set of marriages obtained at the end of the algorithm is stable.

Proof: Assume the contrary.

$\Rightarrow \exists$ marriages (m, w) and (m', w') s.t.

m prefers w' over w and w' prefers m over m'

$\Rightarrow$ m must have proposed to w' before marrying w

$\Rightarrow$ w' likes the man m'' she is married to after the proposal at least as much as m

(either $m'' = m$ or w' rejected m in favour of m'').

Let $m'' = m_1, m_2, \dots, m_t = m'$ be the sequence of partners w' has from this moment forward.

If $t=1$, then $m' = m''$. Since w' likes m'' at least as much as m and $m' \neq m$, w' likes $m'' = m'$ strictly better than m, a contradiction.

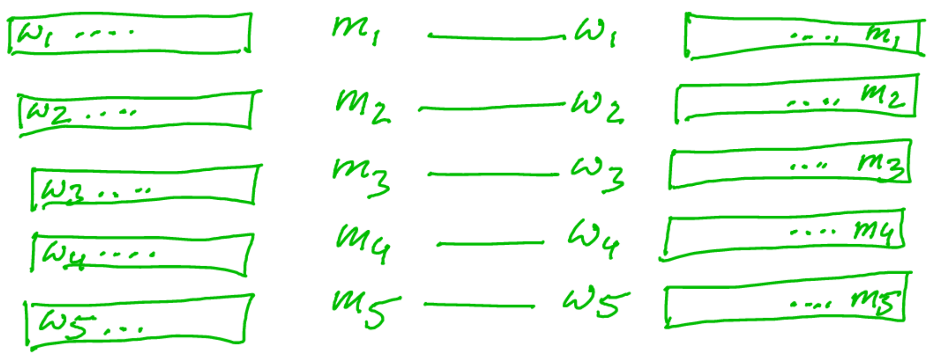
If $t > 1$, then w' likes m_{i+1} strictly better than m_i, for all $1 \leq i < t$ because she divorces m_i for m_{i+1}. Thus, she likes $m' = m_t$ strictly better than $m'' = m_1$, who she likes at least as much as m.

$\Rightarrow$ w' likes m' strictly better than m, again a contradiction. $\square$

More interesting questions

1. Does the solution depend on the order in which the men propose? No!!!

2. Is the solution equally fair to both sexes?
The men fare much better:



3. Can we design a faster algorithm?
Yes, using randomization.

4. How fast can we implement this algorithm?
(Next topic)