ANALYSIS OF ALGORITHMS

LECTURE 11 : STABLE MARRIAGE

BASED ON SECTION 4.3

WHAT IS STABLE MARRIAGE?

- The stable marriage problem is about constructing a set of marriages between men and women such that no marriage ends in divorce
- There are several things that make this diverge from the real world
  - There is an equal number of men and women
  - Every person must be married
  - There is no same-sex marriage
  - We have perfect information about people's preferences

WHAT MAKES THE MARRIAGES STABLE?

- In the real world, divorce can happen because someone is unhappy
- But in this fake world, every person MUST be married
- Under these conditions, why would you get divorced?
- Because you want to marry someone better!
- Is that sufficient though?
  - No! They must also want to marry you, otherwise you'd have no one to marry, and you MUST be married
  - Basically unless you have someone to run away with, you won't run away

WHY?

- Can we compel people to marry whoever we want?
- This is an analogy to many resource allocation problems
- The source problem comes from the problem of assigning interns to hospitals

WHAT INFO DO WE HAVE?

- For every man and woman, we have an ordered list representing their preferences.
- Consider a world with 8 people in it – 4 men, 4 women
- Let's look at man 1
  - He has a list that could be [3,1,2,0]
  - This would mean his favourite woman is woman 3, and his least favourite woman is woman 0
  - We have one of these lists for each of the men and women

PREFERENCE ARRAYS

- These lists are combined into preference arrays
- Looking at this:
  - man 0’s favourite woman is woman 0

Rank

man	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

PREFERENCE ARRAYS

- These lists are combined into preference arrays
- Looking at this:
  - man 0's favourite woman is woman 0
  - man 2's second favourite woman is woman 3

Rank

		0	1	2	3
	0	0	2	1	3
man	**1**	2	0	1	3
	2	2	3	0	1
	3	1	0	3	2

Rank

		0	1	2	3
	0	3	1	2	0
woman	**1**	3	0	2	1
	2	1	3	2	0
	3	1	0	2	3

PREFERENCE ARRAYS

- These lists are combined into preference arrays
- Looking at this:
  - man 0's favourite woman is woman 0
  - man 2's second favourite woman is woman 3
  - woman 1's least favourite man is man 1

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

PREFERENCE ARRAYS

- These lists are combined into preference arrays
- Looking at this:
    - man 0's favourite woman is woman 0
    - man 2's second favourite woman is woman 3
    - woman 1's least favourite man is man 1
    - woman 3's third favourite man is man 2

Rank

		0	1	2	3
man	0	0	2	1	3
	1	2	0	1	3
	2	2	3	0	1
	3	1	0	3	2

Rank

		0	1	2	3
woman	0	3	1	2	0
	1	3	0	2	1
	2	1	3	2	0
	3	1	0	2	3

MARRIAGES

Marriages

M	W
0	0
1	3
2	2
3	1

- When we output a list of marriages, we could output an array stating which person each man (or woman) is married to
- For example in the array presented, man 1 is married to woman 3

Rank

		0	1	2	3
	0	0	2	1	3
man	1	2	0	1	3
	2	2	3	0	1
	3	1	0	3	2

Rank

		0	1	2	3
	0	3	1	2	0
woman	1	3	0	2	1
	2	1	3	2	0
	3	1	0	2	3

STABILITY

Marriages

M	W
0	0
1	3
2	2
3	1

- Is this set of marriages stable?
- How do we check?
- Is a marriage unstable if any of the partners would rather marry someone else?
- No! It’s only unstable if that other person would also rather marry them than their current partner

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

STABILITY

Marriages

M	W
0	0
1	3
2	2
3	1

- So we can check it using the algorithm presented on pg. 52 of the notes. (Please pause the video and go through that algorithm now)

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

STABILITY

Marriages

M	W
0	0
1	3
2	2
3	1

- So we can check it using the algorithm presented on pg. 52 of the notes. (Please pause the video and go through that algorithm now)
- In summary, it says that to check if a marriage between m and w is stable:
  - for every man b that w likes more than her husband m
    - If b likes w more than his wife
      - return false
  - return true

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

STABILITY

Marriages

M	W
0	0
1	3
2	2
3	1

- Then, to check if a whole set of marriages is stable, we can just stick that in a for loop
    - For every woman w
        - for every man b that w likes more than her husband m
            - If b likes w more than his wife
                - return false
    - return true
- (The notes use slightly different terminology. Please read and understand the entire section)

Rank

		0	1	2	3
man	0	0	2	1	3
	1	2	0	1	3
	2	2	3	0	1
	3	1	0	3	2

Rank

		0	1	2	3
woman	0	3	1	2	0
	1	3	0	2	1
	2	1	3	2	0
	3	1	0	2	3

STABILITY

Marriages

M	W
0	0
1	3
2	2
3	1

- Then, to check if a whole set of marriages is stable, we can just stick that in a for loop
  - For every woman w
    - for every man b that w likes more than her husband m
      - If b likes w more than his wife
        - return false
  - return true

- Is the set of marriages on the left stable?

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

STABILITY

Marriages

M	W
0	0
1	3
2	2
3	1

- Then, to check if a whole set of marriages is stable, we can just stick that in a for loop
    - For every woman w
        - for every man b that w likes more than her husband m
            - If b likes w more than his wife
                - return false
    - return true

- Is the set of marriages on the left stable?
- How hard was it to check?

Rank

		0	1	2	3
	0	0	2	1	3
man	1	2	0	1	3
	2	2	3	0	1
	3	1	0	3	2

Rank

		0	1	2	3
	0	3	1	2	0
woman	1	3	0	2	1
	2	1	3	2	0
	3	1	0	2	3

STABILITY

Marriages

M	W
0	0
1	3
2	2
3	1

- Then, to check if a whole set of marriages is stable, we can just stick that in a for loop
  - $\Theta(n)$ For every woman w
    - $\Theta(n)$ for every man b that w likes more than her husband m
      - $\Theta(n)$ If b likes w more than his wife
        - return false
  - return true

- Is the set of marriages on the left stable?
- How hard was it to check?

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

STABILITY

Marriages

M	W
0	0
1	3
2	2
3	1

- Then, to check if a whole set of marriages is stable, we can just stick that in a for loop
  - $\Theta(n)$ For every woman w
    - $\Theta(n)$ for every man b that w likes more than her husband m
      - $\Theta(n)$ If b likes w more than his wife
        - return false
  - return true

- Is the set of marriages on the left stable?
- How hard was it to check? $\Theta(n^3)$

		Rank			
		0	1	2	3
man	0	0	2	1	3
	1	2	0	1	3
	2	2	3	0	1
	3	1	0	3	2

		Rank			
		0	1	2	3
woman	0	3	1	2	0
	1	3	0	2	1
	2	1	3	2	0
	3	1	0	2	3

STABILITY

Marriages

M	W
0	0
1	3
2	2
3	1

- $\Theta(n^3)$ just to check if a set of marriages is stable? Argh!
- We can do better!
- Where is the piece we can improve? What part of that cost is based on our decisions rather than the problem itself?
- It's that linear search for whether b likes w more than his wife…
- What if we instead store how much b likes each girl? Then we could just look it up.

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man

	0	1	2	3
0				
1				
2				
3				

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man

	0	1	2	3
0	0			
1				
2				
3				

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man

	0	1	2	3
0	0	2		
1				
2				
3				

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man	0	1	2	3
0	0	2	1	
1				
2				
3				

Rank

man	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man

	0	1	2	3
0	0	2	1	3
1				
2				
3				

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man

	0	1	2	3
0	0	2	1	3
1	1			
2				
3				

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man

	0	1	2	3
0	0	2	1	3
1	1	2		
2				
3				

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man

	0	1	2	3
0	0	2	1	3
1	1	2	0	
2				
3				

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man

	0	1	2	3
0	0	2	1	3
1	1	2	0	3
2				
3				

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man

	0	1	2	3
0	0	2	1	3
1	1	2	0	3
2	2	3	0	1
3	1	0	3	2

Rank

man

	0	1	2	3
0	0	2	1	3
1	2	0	1	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	**W**
0	0
1	3
2	2
3	1

- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman

man

	0	**1**	**2**	**3**
0	0	2	1	3
1	1	2	0	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	**1**	**2**	**3**
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- How does that affect the complexity?
  - For every woman w
    - for every man b that w likes more than her husband m
      - If b likes w more than his wife
        - return false
  - return true

woman

man

	0	1	2	3
0	0	2	1	3
1	1	2	0	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- How does that affect the complexity?

$\Theta(n)$
- For every woman w

$\Theta(n)$
  - for every man b that w likes more than her husband m
    - If b likes w more than his wife
      - return false
- return true

woman

man

	0	1	2	3
0	0	2	1	3
1	1	2	0	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3

RANKING ARRAYS

Marriages

M	W
0	0
1	3
2	2
3	1

- How does that affect the complexity?
  - $\Theta(n)$ For every woman w
    - $\Theta(n)$ for every man b that w likes more than her husband m
      - $\Theta(1)$ If b likes w more than his wife
        - return false
  - return true
- Total complexity is $\Theta(n^2)$

woman

man

	0	1	2	3
0	0	2	1	3
1	1	2	0	3
2	2	3	0	1
3	1	0	3	2

Rank

woman

	0	1	2	3
0	3	1	2	0
1	3	0	2	1
2	1	3	2	0
3	1	0	2	3