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 I
 - 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

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

Rank								
		0	I	2	3			
man	0	0	2	I	3			
	I	2	0	ı	3			
	2	2	3	0	ı			
	3	I	0	3	2			
			Ran	k				
		0	1	2	3			
	0	3	I	2	0			
voman	1	3	0	2	ı			
	2	I	3	2	0			
	3	I	0	2	3			

- 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

			Ran	k	
		0	1	2	3
man	0	0	2	I	3
	1	2	0	I	3
	2	2	3	0	I
	3	Ī	0	3	2
			Ran	k	
		0	1	2	3
	0	3	I	2	0
woman	1	3	0	2	I
	2	I	3	2	0
	3	I	0	2	3

- 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 I's least favourite man is man I

Rank								
		0	I	2	3			
man	0	0	2	I	3			
		2	0	ı	3			
	2	2	3	0	ı			
	3	I	0	3	2			
			Ran	k				
		0	1	2	3			
	0	3	I	2	0			
voman	I	3	0	2	1			
	2	I	3	2	0			
	3	I	0	2	3			

- 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 I's least favourite man is man I
 - woman 3's third favourite man is man 2

Rank									
man		0	1	2	3				
	0	0	2	1	3				
	1	2	0	I	3				
	2	2	3	0	I				
	3	ı	0	3	2				
			Ran	k					
		0	I	2	3				
	0	3	I	2	0				
voman		3	0	2	I				
	2	I	3	2	0				
	3	I	0	2	3				

MARRIAGES

Marriages

- 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 I is married to woman 3

	Rank								
		0	I	2	3				
	0	0	2	I	3				
man	1	2	0	1	3				
	2	2	3	0	ı				
	3	I	0	3	2				
			Ran	k					

	Tain							
		0	1	2	3			
	0	3	1	2	0			
woman	I	3	0	2	Ī			
	2	I	3	2	0			
	3	ı	0	2	3			

Marriages

M W

. .

. .

3

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

			Ran	k	
		0	I	2	3
	0	0	2	I	3
man	1	2	0	ĺ	3
	2	2	3	0	1
	3	I	0	3	2
			_	-	

	Rank									
		0	1	2	3					
	0	3	I	2	0					
voman	I	3	0	2	l					
	2	I	3	2	0					
	3	I	0	2	3					

Marriages

 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)

			Ran	<u> </u>	
		0	1	2	3
	0	0	2	1	3
man	1	2	0	I	3
	2	2	3	0	I
	3	ı	0	3	2
			_	_	
			Ran	k	
		0	Ran	2	3
	0	0	Ran I		3
woman	0 I		1	2	
woman		3	I I	2	0

Marriages

MW0132231

- 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									
		0	1	2	3				
man	0	0	2	ı	3				
	1	2	0	ı	3				
	2	2	3	0	I				
	3	I	0	3	2				
			Ran	k					
		0	1	2	3				
	0	3	I	2	0				
woman	I	3	0	2	I				
	2	Ī	3	2	0				
	3	I	0	2	3				

Marriages

MOI3

• 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	I	2	3			
	0	0	2	I	3			
man		2	0	Ī	3			
	2	2	3	0	1			
	3	I	0	3	2			
,			Ran	k				

	Kank								
		0	1	2	3				
	0	3	I	2	0				
woman	I	3	0	2	I				
	2	I	3	2	0				
	3	I	0	2	3				

Marriages

MW00

3

2 2

3 I

• 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								

	Rank					
		0	1	2	3	
	0	3	I	2	0	
woman	I	3	0	2	I	
	2	ı	3	2	0	
	3	1	0	2	3	

Marriages

 M
 W

 0
 0

2 2

3 I

• 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	I	2	3		
	0	0	2	I	3		
man	1	2	0	1	3		
	2	2	3	0	I		
	3	Ī	0	3	2		

	Rank					
		0	1	2	3	
	0	3	I	2	0	
woman	1	3	0	2	ı	
	2	I	3	2	0	
	3	I	0	2	3	

Marriages

 $\Theta(n)$ $\Theta(n)$

 $\Theta(n)$

• 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	I	2	3		
	0	0	2	I	3		
man	1	2	0	1	3		
	2	2	3	0	1		
	3	I	0	3	2		
			_				

	Kank				
		0	I	2	3
	0	3	I	2	0
woman	I	3	0	2	I
	2	I	3	2	0
	3	I	0	2	3

Marriages

M0

just stick that in a for loop

1 3

 $\Theta(n)$ •

 $\Theta(n)$

2 2

 $\Theta(n)$

• For every woman w

• for every man b that w likes more than her husband m

• Then, to check if a whole set of marriages is stable, we can

- 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				
	0	0	2	I	3				
man		2	0	Ī	3				
	2	2	3	0	1				
	3	1	0	3	2				

	Kank					
		0	1	2	3	
	0	3	I	2	0	
woman	I	3	0	2	I	
	2	I	3	2	0	
	3	I	0	2	3	

Marriages

MW

0 0

3

2 2

3

• $\Theta(n^3)$ just to check if a set of marriages is stable? Argh!

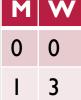
- We can do better!
- Where is the piece we can improve? What part of that cost is based on our decisions rather than the problem itself?
- It's that linear search for whether b likes w more than his wife...
- What if we instead store how much b likes each girl? Then we could just look it up.

Rank								
	0	I	2	3				
0	0	2	I	3				
	2	0	Ī	3				
2	2	3	0	I				
3	I	0	3	2				
	0 I 2	0 0 I 2	0 I 0 0 2 1 2 0 2 2 3	0 I 2 0 0 2 I I 2 0 I 2 2 3 0	0 I 2 3 0 0 2 I 3 I 2 0 I 3 2 2 3 0 I			

m

	Rank				
		0	1	2	3
	0	3	I	2	0
woman	1	3	0	2	ı
	2	1	3	2	0
	3	Ī	0	2	3

Marriages



2 2

 A ranking array stores the same information as a preference array, but organized differently

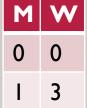
Axes are man and woman, and item is rank

 Let's convert that preference array for men into a ranking array
 woman

	Kank							
		0	I	2	3			
man	0	0	2	I	3			
	1	2	0	I	3			
	2	2	3	0	ı			
	3	I	0	3	2			
	 Rank							

	Kalik					
		0	1	2	3	
	0	3	I	2	0	
woman	1	3	0	2	I	
	2	1	3	2	0	
	3	Ī	0	2	3	

Marriages



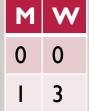
- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array
 woman

		0	I	2	3
	0	0			
man					
	2				
	3				

	Kank						
		0	I	2	3		
	0	0	2	I	3		
man		2	0	l	3		
	2	2	3	0	1		
	3	I	0	3	2		
			Ran	k			

			ixaii	<u> </u>	
		0	1	2	3
	0	3	1	2	0
woman	1	3	0	2	I
	2	I	3	2	0
	3	I	0	2	3

Marriages



 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

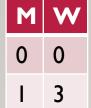
		0	I	2	3
man	0	0	2		
	1				
	2				
	3				

	Kank						
		0	1	2	3		
	0	0	2	I	3		
man	1	2	0	ı	3		
	2	2	3	0	I		
	3	I	0	3	2		
			Ran	ما			

Dank

	Nank O I 2 3 O 3 I 2 0 I 3 0 2 I 2 I 3 2 0 3 I 0 2 3				
		0	1	2	3
	0	3	I	2	0
woman	1	3	0	2	l
	2	I	3	2	0
	3	I	0	2	3

Marriages



- 2 2
- 3

- 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

		0	1	2	3
	0	0	2	I	
man					
	2				
	3				

	Kank						
		0	1	2	3		
	0	0	2	I	3		
man	1	2	0	I	3		
	2	2	3	0	ı		
	3	I	0	3	2		
		•			_		

			Kan	K	
		0	1	2	3
	0	3	I	2	0
woman	I	3	0	2	l
	2	I	3	2	0
	3	I	0	2	3

Marriages

MW0I3

2 2

3

 A ranking array stores the same information as a preference array, but organized differently

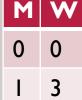
Axes are man and woman, and item is rank

 Let's convert that preference array for men into a ranking array
 woman

	Kank						
		0	I	2	3		
	0	0	2	I	3		
man	1	2	0	1	3		
	2	2	3	0	I		
	3	I	0	3	2		
			Ran	ما			

		Kank						
		0	1	2	3			
	0	3	I	2	0			
woman	I	3	0	2	l			
	2	I	3	2	0			
	3	I	0	2	3			

Marriages



 A ranking array stores the same information as a preference array, but organized differently

• Axes are man and woman, and item is rank

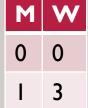
 Let's convert that preference array for men into a ranking array
 woman

		0	I	2	3
	0	0	2	I	3
man	I	I			
	2				
	3				

	Rank						
		0	1	2	3		
	0	0	2	I	3		
man	1	2	0	1	3		
	2	2	3	0	ı		
	3	I	0	3	2		
			Ran	L			

	Nalik						
		0	1	2	3		
	0	3	I	2	0		
woman	1	3	0	2	l		
	2	I	3	2	0		
	3	I	0	2	3		

Marriages



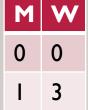
- A ranking array stores the same information as a preference array, but organized differently
- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array
 woman

		0	I	2	3
	0	0	2	I	3
man	I	I	2		
	2				
	3				

	Kank					
		0	1	2	3	
	0	0	2	I	3	
man	1	2	0	l	3	
man 1 2	2	2	3	0	I	
	3	I	0	3	2	
			Ran	L		

		Kank						
		0	1	2	3			
	0	3	I	2	0			
woman	1	3	0	2	l			
	2	I	3	2	0			
	3	I	0	2	3			

Marriages



- 2 2
- 3

- 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

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

	Kank					
		0	1	2	3	
	0	0	2	I	3	
man		2	0	l	3	
	2	2	3	0	1	
	3	I	0	3	2	

	Kank						
		0	1	2	3		
	0	3	I	2	0		
woman	I	3	0	2	I		
	2	I	3	2	0		
	3	I	0	2	3		

Marriages



I 3

- 2 2
- 3

 A ranking array stores the same information as a preference array, but organized differently

• Axes are man and woman, and item is rank

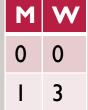
 Let's convert that preference array for men into a ranking array
 woman

		0	I	2	3
	0	0	2	I	3
man	I	I	2	0	3
	2				
	3				

	Kank						
		0	1	2	3		
	0	0	2	I	3		
man		2	0	l	3		
	2	2	3	0	1		
	3	I	0	3	2		
			_				

		Kank						
		0	1	2	3			
	0	3	I	2	0			
woman	1	3	0	2	I			
	2	1	3	2	0			
	3	I	0	2	3			

Marriages



- 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	I	3
	I	I	2	0	3
	2	2	3	0	ı
	3	I	0	3	2

	Kank						
		0	1	2	3		
	0	0	2	I	3		
man	1	2	0	l	3		
	2	2	3	0	I		
	3	I	0	3	2		
Rank							

	Kank						
		0	1	2	3		
	0	3	I	2	0		
woman	I	3	0	2	I		
	2	1	3	2	0		
	3	Ī	0	2	3		

Marriages

 M
 W

 0
 0

1 3

2 2

3

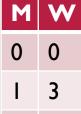
• A ranking array stores the same information as a preference array, but organized differently

- Axes are man and woman, and item is rank
- Let's convert that preference array for men into a ranking array

woman								
		0	I	2	3			
nan	0	0	2	I	3			
	1	I	2	0	3			
	2	2	3	0	I			
	3	I	0	3	2			
Rank								
		0	1	2	3			
nan	0	3	I	2	0			
	I	3	0	2	I			
	2	I	3	2	0			
	3	I	0	2	3			

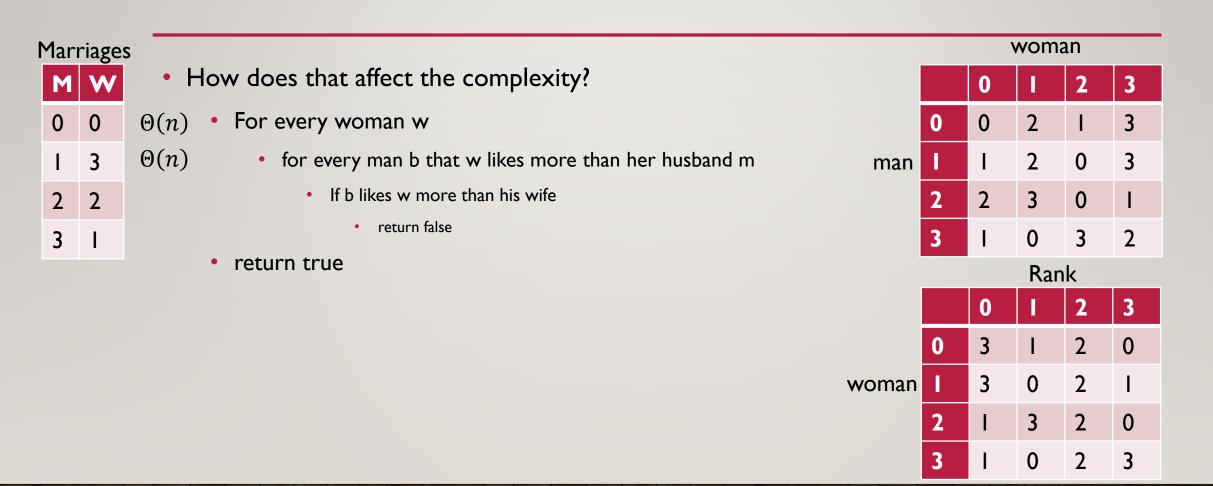
wom

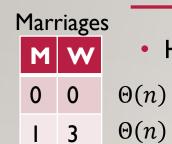
Marriages



- 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								
		0	1	2	3			
man	0	0	2	1	3			
	1	I	2	0	3			
	2	2	3	0	I			
	3	Ī	0	3	2			
Rank								
oman		0	1	2	3			
	0	3	I	2	0			
	I	3	0	2	I			
	2	I	3	2	0			
	3	ı	0	2	3			





 $\Theta(1)$

How does that affect the complexity?

 $\Theta(n)$ • 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

• Total complexity is $\Theta(n^2)$

woman									
		0	I	2	3				
man	0	0	2	I	3				
	I	I	2	0	3				
	2	2	3	0	I				
	3	ı	0	3	2				
	Rank								
voman		0	I	2	3				
	0	3	I	2	0				
	I	3	0	2	Ī				
	2	I	3	2	0				
	3	I	0	2	3				