Online Tools and Sources of Information

D2L:

Pre-Lecture Notes

Lecture Notes

HW Assignments Posted

Syllabus

Links to Piazza and Office Hours Schedule

Any other documnet shared during the semester

Lecture videos and Discussion videos

Piazza:

Discussion Board

Zoom:

Some Office Hours

Grade Scope:

HW submissions

Exams

Discussion Sessions Fridays 11-1 @ OHE 100B

Roles and Responsibilities

Instructor	Lectures
TAs	HW, Exams, Office hrs
Course Producers	HW grading, some office hrs
CS Dept Advisors	registration issues
DEN Support	tech issues

Textbook

Algorithm Design by John Kleinberg and Eva Tardos

Supplemental Textbook

Introduction to Algorithms, 3rd Edition., by Cormen et al.

Your Responsibilities

Attend Lectures and Discussion Sessions

Complete the reading assignments from textbook

Complete and submit HW assignments

Do as many other practice problems as possible

Exams

Your Grade

Exam 1	*45 %*	July 24
Exam 2	*45 %*	Aug. 7
HW assignments	*10%*	
	100%	

Grading Scale

90 - 100	*A*	*60 - 64.99*	*C+*
86 - 89.99	*A-*	*55 - 59.99*	*C*
80 - 85.99	*B+*	*50 - 54.99*	*C-*
70 - 79.99	*B*	*45 - 49.99*	*D*
65 - 69.99	*B-*	*Below 45*	*F*

Adjustments to the Grading Scale

Scale will be adjusted if the overall class average falls below 75

At least the top 20% of the class will receive an A

At least the next 10% of the class will receive an A-

Prerequisites

Discrete Math (Mathematical Induction)

Sorting Methods

Basic Data Structures:

- Arrays, Linked Lists, Stack, Queue

Graphs Basics:

- Tree, Path, Cycle, Directed/Undirected, DAG, Adjacency List/Matrix

Graphs Search Algorithms:

- BFS, DFS

High level Syllabus

Today!

- Introduction

Tomorrow

- Review of some prereq's + asymptotic notation
- Major algorithmic techniques
  - Greedy
  - Divide & Conquer
  - Dynamic Programming
- Network Flow
- Computational Complexity Theory
- Approximation Algorithms
- Linear Programming

Corrections

1- ~~An algorithm is a set of instructions in machine language.~~

Kharazmi 780-850

2- ... ~~Algorithmic science advanced on Wall Street ...~~

3- ... ~~Invite 6 million algorithms for a listen ...~~

Lecture tomorrow @ 9:00AM

in OHE 100D

When studying a problem, we go through the following steps:

1- Come up with a concise problem statement

2- Present a solution

3- Prove correctness

4- Perform complexity analysis

Stable Matching

Stable Matching Example

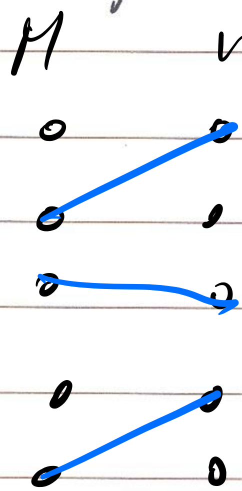
Problem: We are interested in matching $\underline{n}$ men with $\underline{n}$ women so that they could stay happily married ever after.

Step 1: Come up with a concise problem statement.

We have a set of $\underline{n}$ men, $M = \{m_1, \dots m_n\}$

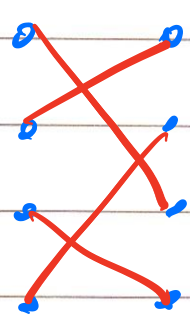
We have a set of $\underline{n}$ women, $W = \{w_1, \dots w_n\}$

<u>Def.</u> A <u>Matching</u> S is a set of ordered pairs.



A matching

<u>Def.</u> A <u>perfect matching</u> S' is a matching with the property that each member of M and each member of W appear in <u>exactly one pair</u> in S'.



A perfect Matching

Add notion of preferences

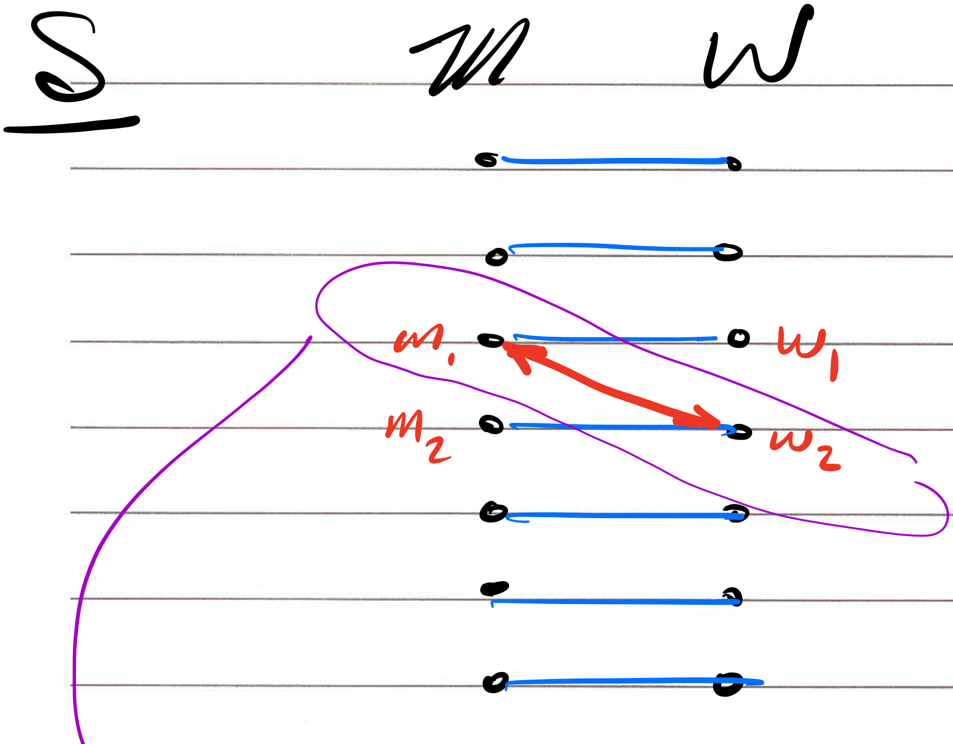
Each man $m \in M$ ranks all women

- $\underline{m}$ prefers $\underline{w}$ to $\underline{w'}$ if $\underline{m}$ ranks $\underline{w}$ higher than w'.

- Ordered ranking of $\underline{m}$ is his preference list

$$P_{m_i} = \{ W_{i_1}, W_{i_2}, \ldots W_{i_n} \}$$

Same for women, i.e. each woman $w \in W$ ranks all men ...



Such a pair is called an instability

WRT S

Def. Matching S is stable if

1- It is perfect

2- There are no instabilities WRT S

Step 1: Input: Preference lists for a set of n men & n women.

Output: Set of n marriages w/ no instabilities

Step 2: Gale-Shapley Alg.

Step 3 Proof of Correctness

① From the woman's perspective, she starts single, and once she gets engaged she can only get into "better" engagements

② From the man's perspective, he starts single, gets engaged, but might get dropped repeatedly, only to settle for a lower ranking woman.

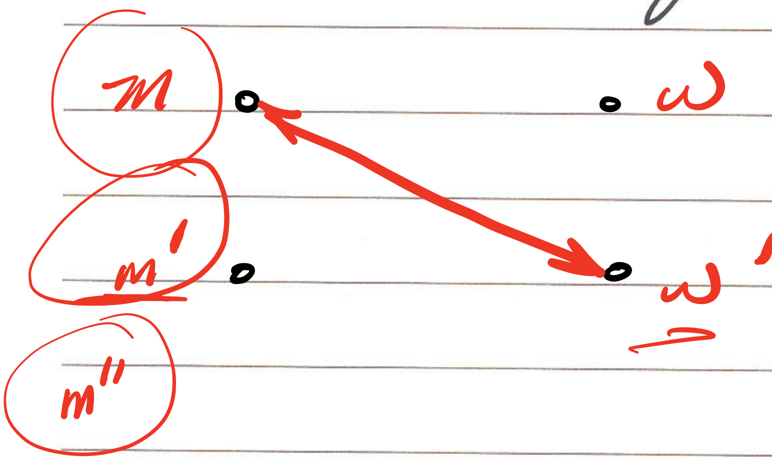
③ Algorithm terminates in at most n^2 iterations

④ The solution will be a perfect matching

⑤ Need to show that the sol. is a stable matching

Proof by Contradiction

Assume an instability exists in our solution involving two pairs $(m, w), (m', w')$

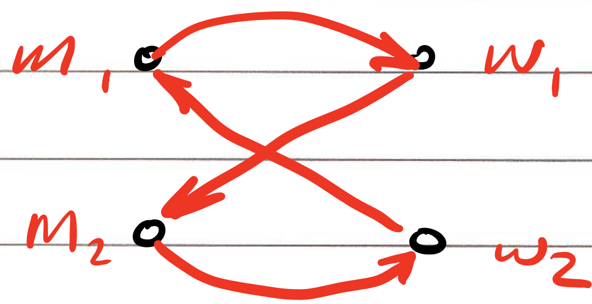


Q: Did $\underline{m}$ propose to w' at some point in the execution?

If no, then $\underline{w}$ must be higher than $\underline{w'}$ on his list $\Rightarrow$ contradiction!

If yes, he must have been rejected in favor of m'' and due to ① either $m'' = m'$ or m' is better than m''

$\Rightarrow$ Contradiction!



Men proposing $\longrightarrow$ $(m_1, w_1), (m_2, w_2)$

Women " $\longrightarrow$ $(m_1, w_2), (m_2, w_1)$

Step 4 Complexity Analysis

✓ 1- Identify a free man

✓ 2- For a man $\underline{m}$, identify the highest ranked woman to whom he has not yet proposed.

✓ 3- For a woman $\underline{w}$, decide if $\underline{w}$ is engaged, and if so to whom

4- For a woman $\underline{w}$ and two men $\underline{m}$ & $\underline{m'}$, decide which man is preferred by $\underline{w}$

✓ 5- Place a man back in the list of free men.

1. Identify a free man

	get	put
array	O(1)	O(1)
queue	O(1)	O(1)
stack	O(1)	O(1)
linked list	O(1)	O(1)

2. Identify the highest ranked woman to whom m has not yet proposed.

Keep an array Next [1..n], where Next [m] points to the position of the next woman he will be proposing to on his pref. list.

Men's preference list: ManPref [1..n, 1..n], where

ManPref [m, i] denotes the i^{th} woman on man m's preference list.

To find next woman w to whom m will be proposing to:

$$w = \underbrace{ManPref[m, Next[m]]}_{\text{takes } O(1)}$$

3_ Determine woman w's status

Keep an array called Current[1..n] where Current[w] is Null if w is single and set to m if w is engaged to m.

takes O(1)

4_ Determine which man is preferred by w.

$WomanPref_i$

1	2	3	4	5				
3	8	4	32	1	.	. .	.	

$WomanRanking_i$

1	2	3	4					
5		1	3					

Preparation before entering GS iterations

Create a Ranking array where Ranking [w,m] contains the rank of man m based on w's preference

Preparation + GS iterations

$O(n^2)$ $O(n^2)$

overall complexity $= O(n^2)$