Platforms & Tools

D2L (DEN) : Syllabus
Lecture Notes
Lecture Videos
HW Assignments
HW Submissions
Any other reference material
online Exams

Piazza : Discussion Board

Roles & Responsibilities

- Instructor   lectures & Discussions
- TAs   HWs & Exams
- Graders   HW grading
- Course Producers   *
- CS Dept. Advisors
- DEN Support

Textbooks

- Algorithms Design by Jon Kleinberg & Eva Tardos

- Supplemental Textbook:
  Introduction to Algorithms, 3rd edition, by Cormen et al.

Your Responsibilities

1 - Attending lectures & Discussions
2 - Completing reading assignments
3 - Doing HW problems
4 - Doing as many other problems from textbook as possible

Your Grade

Exam 1	30%	Oct. 1
Exam 2	30%	Nov. 5
Exam 3	35%	Dec. 3
Final Proj	5%	Dec 8
	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 45.99	F

- Scale will be adjusted if median falls below 75.
- At least the top 20% of the class will receive an A.
- At least the next 10% of the class (between top 20% and top 30%) will receive an A-.

Prerequisites

- Discrete Math – Mathematical Induction
- Sorting methods
- Basic data structures: Arrays, stacks, queues, linked lists
- Basics of graphs: Trees, cycles, DAG, adjacency list/matrix, etc.
- Graph search algorithms: BFS, DFS

High level Syllabus

Today!

- Introduction
- Review of some prereq's + asymptotic notation
- Major algorithmic techniques
  - Greedy
  - Divide & Conquer
  - Dynamic Programming

Exam 1

- Network Flow
- Computational Complexity Theory
- Approximation Algorithms
- Linear Programming

Exam 2

covers only 2 topics

Exam 3

comprehensive

Corrections

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

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

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

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

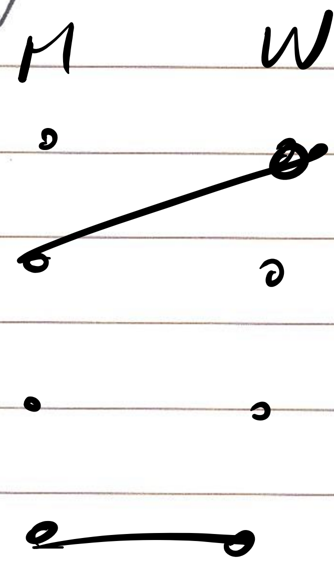
Problem: We are interested in matching $\underline{n}$ men with $\underline{n}$ women so that they could stay happily married ever after.

Step 1: Come up with a concise problem statement.

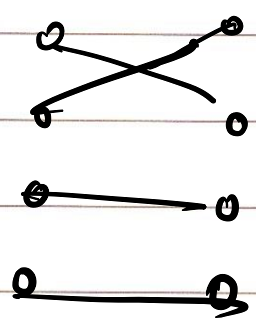
We have a set of $\underline{n}$ men, $M = \{m_1, \dots m_n\}$

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

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



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



A perfect matching

Add notion of preferences

Each man $m \in M$ ranks all women

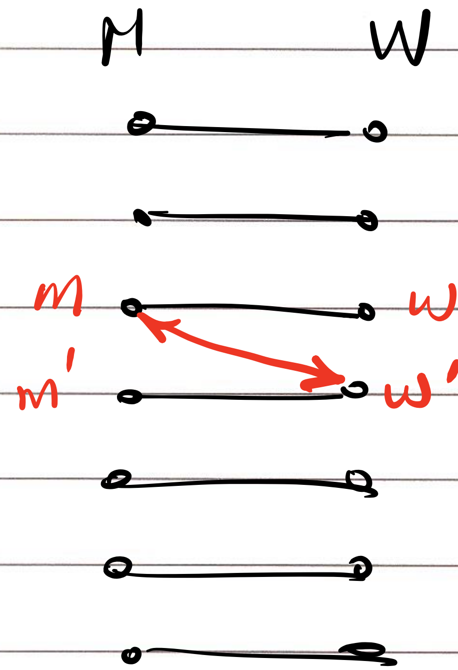
- $\underline{m}$ prefers $\underline{w}$ to $\underline{w'}$ if $\underline{m}$ ranks $\underline{w}$ higher than w'.

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

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

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

S



Such a pair (m, w') is called an instability

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 and she can only get into better engagements

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

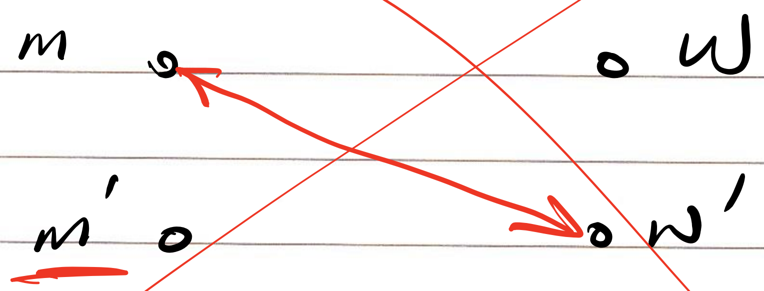
3. Solution will terminate in at most n^2 iterations

4. Solution is a perfect matching

5. Solution is a stable "

Proof by Contradiction

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



Say (m, w') is an instability

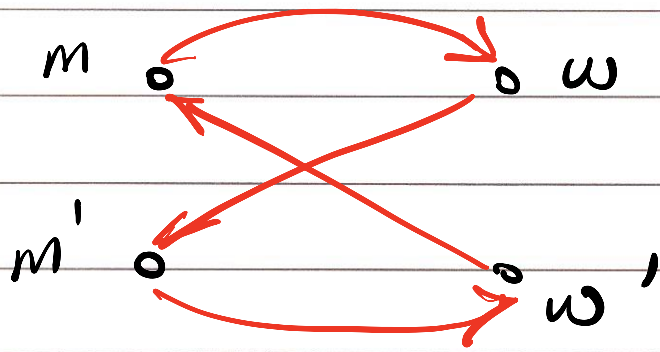
m'', m'''

Q: Did m propose to w' at some point in the execution?

If no, then w must be higher than 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!



$(m, w), (m', w')$

$(m, w'), (m', w)$

Step 4 Complexity Analysis

1- Identify a free man $O(1)$

2- For a man $\underline{m}$, identify the highest ranked woman to whom he has not yet proposed. $O(1)$

3- For a woman $\underline{w}$, decide if $\underline{w}$ is engaged, and if so to whom $O(1)$

4- For a woman $\underline{w}$ and two men $\underline{m}$ & $\underline{m'}$, decide which man is preferred by $\underline{w}$ $O(1)$

5- Place a man back in the list of free men. $O(1)$

1. Identify a free man

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

2. Identify the highest ranked woman to whom <u>m</u> 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 $\underline{m}$'s preference list.

To find next woman $\underline{w}$ to whom $\underline{m}$ will be proposing to:

$$\underline{w} = ManPref[m, Next[m]]$$

takes $O(1)$

3_ Determine woman $\underline{w}$'s status

Keep an array called Current[1..n]
where Current[w] is Null
if w is single & set to $\underline{m}$
if $\underline{w}$ is engaged to $\underline{m}$.

takes O(1)

4_ Determine which man is preferred by $\underline{W}$.

$WomanPref_i$

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

Woman Ranking

5		1	3			
1						

Preparation before entering GS iterations

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

$$\underbrace{\text{Preparation}}_{O(n^2)} + \underbrace{\text{GS iterations}}_{O(n^2)}$$

Overall Complexity $= O(n^2)$

suppose W_{i2}, W_{i4}, W_{i23} are valid partners of M_i

highest

M_i's best valid partner

$$M_i \quad \begin{cases} W_{i1} \\ W_{i2} \\ W_{i3} \\ W_{i4} \\ W_{i23} \\ W_{in} \end{cases}$$

lowest

Def. Woman w is a valid partner of a man m if there is a stable matching that contains the pair (m, w)

not just according to Gale shapely but according to any stable matching algorithm

Def. m's best valid partner is highest rated women, he was paired with in some stable matching

There can be many diff stable matching

Gale shapely algo gives one of those stable matching

Gale shapely always gives same stable matching

Claim: Every execution of the G-S algorithm (When men propose) results in the same stable matching regardless of the order in which men propose.

Plan: to prove this, we will show that when men propose, they always end up with their best valid partner.

Proof by contradiction:

Say <u>m</u> is the first man rejected by a valid partner <u>w</u>. in favor of m'

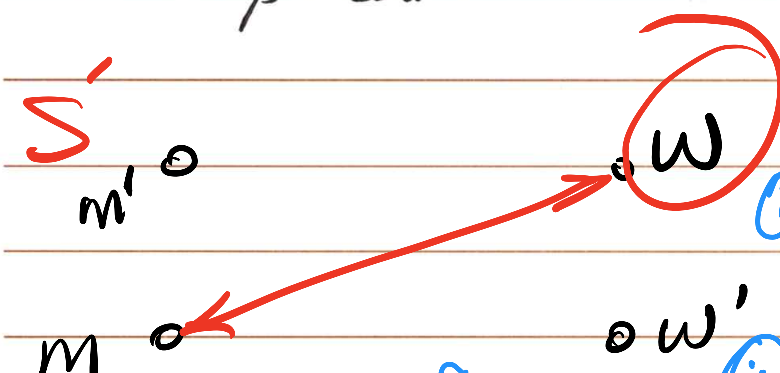
S'

Claim: When men propose, women end up with their worst valid partners

Proof: By contradiction . . .

Suppose we end up with a matching S where for a pair (m, w) in S, m is **not** w's worst valid partner.

So there must be another matching S' where w is paired with a man m' whom she likes less.



If she likes m' less then

(i) If m propose w, they will form pair

(ii) m didnot propose w?
not possible as w is his best valid partner

Hence contradiction