Online Sources of Information

External Class Website:
- Contact information
- Syllabus
- office hours
- exam dates
- policies

DEN Website:
- lecture videos
- lecture notes
- HW assignments
- any other documents

GMT-08

Mon	Tue	Wed	Thu	Fri
			Kuang Liu 8 – 10am SSC 300	
David Kim 10am – 12pm RTH 406	Eunkyung Kim 10am – 12pm SAL 125	KnagGeon Kim 10am – 1pm PHE 101	Ritesh Ahuja 10am – 2pm RTH 323	Chao Yang 10am – 12pm SAL 125
KangGeon Kim 12pm, PHE 101	Eunkyung Kim 1 – 3pm SAL 125	Soheil Alavi 1 – 3pm SAL 125	Lecture 2 6:55 – 9:30pm SGM 123	Dong Guo 2:30 – 6:30pm EEB 220
Amirhossein Mohajeri 2 – 4pm EEB 512	Pengda Xiang 3 – 7pm SAL 125	Soheil Alavi 4 – 6pm SAL 125		
Mohammad Asghari 4 – 6pm RTH 323		Lecture 1 6:40 – 9:20pm SGM 123		

Roles and Responsibilities

- Instructors — Lectures, Discussions
- TAs — HWs & Exams
- Course producers — ?
- Graders — grading of HWs
- CS Dept advisers — registration issues
- DEN — DEN support

Textbooks

- Algorithm Design by Jon Kleinberg & Eva Tardos
- Supplemental textbook: Introduction to Algorithms, 3rd edition by Cormen

Your Responsibilities

- Attend lectures and discussion sessions
- study the material from textbook
- Do HW problems
- Do as many other problems from the textbook as possible

Your grade

Exam 1	30%	Feb 16
Exam 2	30%	March 30
Exam 3	40%	Apr. 27

Prerequisites

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

High Level Syllabus

- Introduction
- Review of some prereq's
- Data structures (Heaps)
- Major algorithmic techniques
  - Greedy
  - Divide & Conquer → Exam 1
  - Dynamic Programming

- Network Flow → Exam 2
- Randomization
- NP, NP-Complete, NP-hard
- Approximation methods
- Linear Programming

→ Exam 3

Corrections:

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

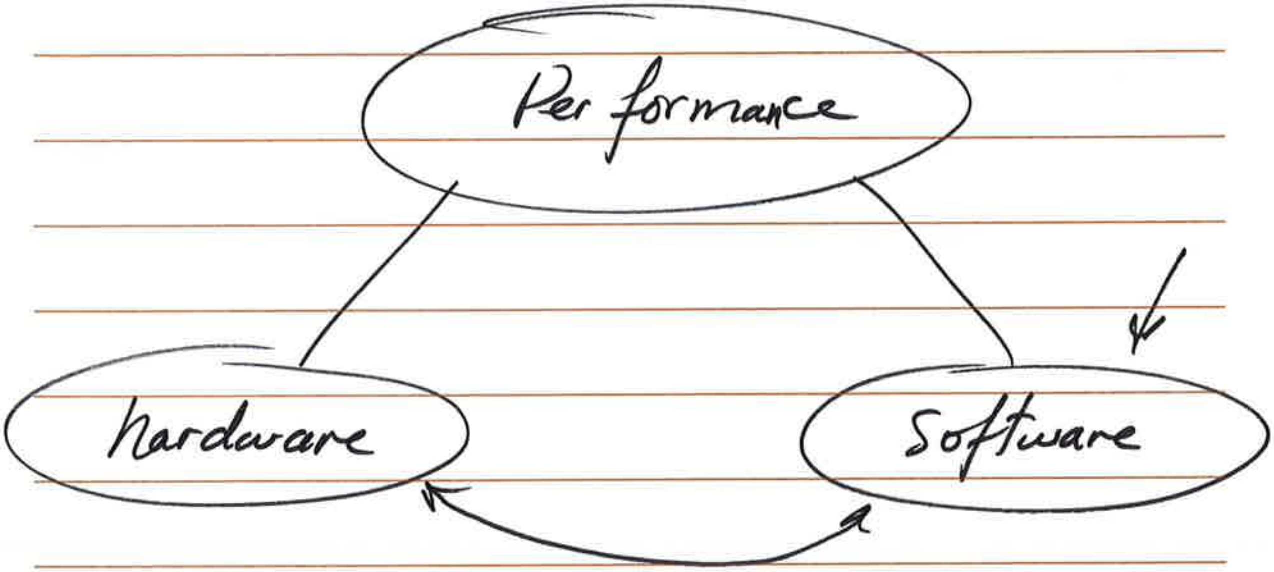
Kharazmi 780-850

Algorithm

2- ~~Alg. science advanced on Wall st...~~

3- ~~... Invite 6 million alg's for a listen...~~

1_ Correctness
2_ Performance



FPGA
Cache
GPU
SMP
DMP

In 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

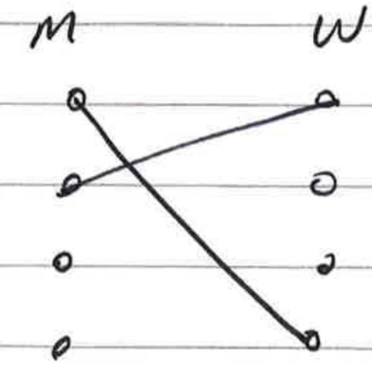
Problem: We are interested in matching $\underline{n}$ men with $\underline{n}$ women so that they could stay happily married ever after.

Step 1: Come up with a concise problem statement.

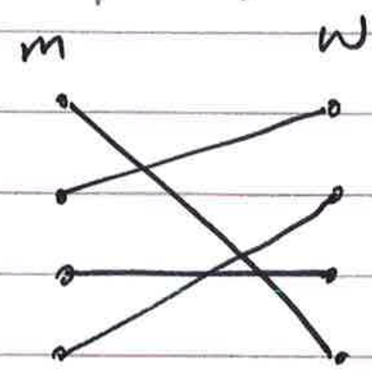
We have a set of $\underline{n}$ men $M = \{m_1, \ldots, m_n\}$

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

Def. A *matching* S is a set of ordered pairs.



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



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 $\underline{w'}$.

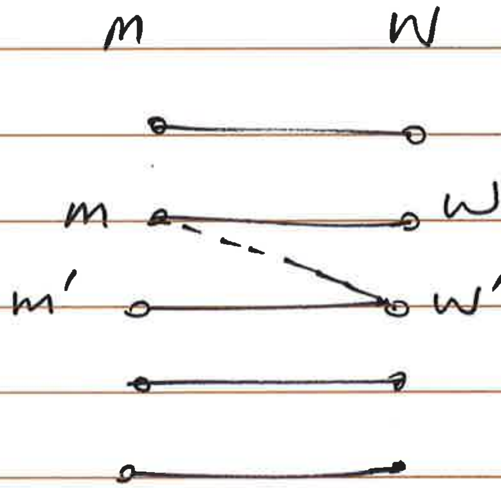
- ordered ranking of $\underline{m}$ is his preference list

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

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

Not prepared!

S



<u>Def.</u> Such a pair (m, w') is an instability WRT S.

proposed!

Def. Matching S is stable if

1- It is perfect

2- There are no instabilities WRT S

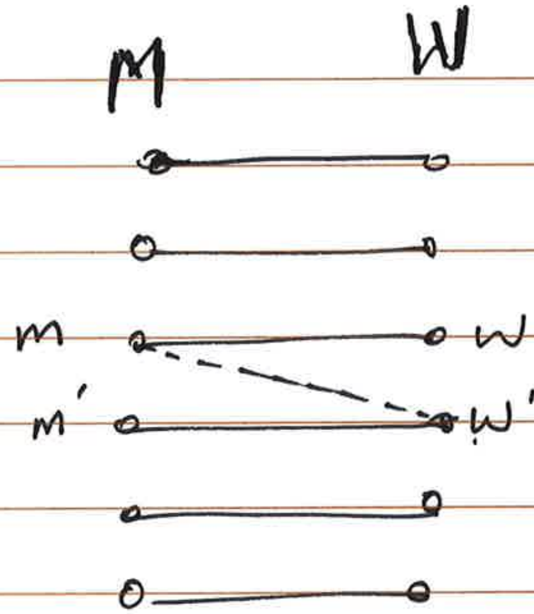
Step 1: Input: Preference lists for a set of n men & n women.

Output: Set of n marriages w/ no instabilities

Step 2 Gale-Shapley Alg.

Did we use any code?

S



Def. Such a pair (m, w') is an *instability* WRT S

Def. Matching S is stable if

1- It is perfect

2- There are no enstabilities WRT S

Complete Step 1: Input: Preference lists for a set of $\underline{n}$ men & n women.

Output: Set of "marriages" w/ no instabilities

Step 2: Gale-Shapley

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, and might be dropped repeatedly only to settle for a woman w/ lower rankings

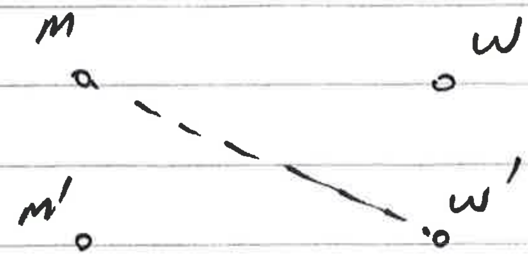
3. Alg. terminates after n^2 iterations

4. Solution is a perfect matching

5. " " a stable "

Proof by Contradiction

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



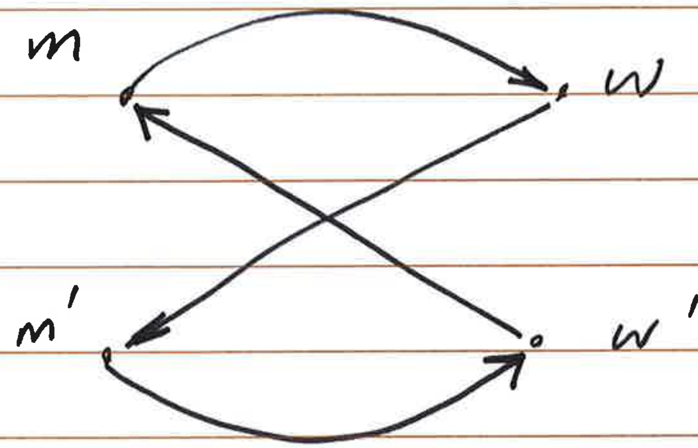
m''

m'''

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

If no, then w must be higher than w' on his list $\Rightarrow$ Contradiction!

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



$(m,w), (m',w')$ men proposing

$(m,w'), (m',w)$ women 〃

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 m & 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

use an array $O(1)$
" a linked list $O(1)$
" a stack $O(1)$
" a queue $O(1)$

Step 1 & 5 will take $O(1)$

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 ranked list.

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

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

To find next woman <u>w</u> to whom <u>m</u> will be proposing to:

$$w = ManPref[m, Next[m]]$$

$O(1)$

3. Decide if <u>w</u> is engaged, and if so to whom.

Keep an array called current[1..n] where

Current[w] is Null if w is not engaged.

" is set to <u>m</u> if she is engaged to <u>m</u>.

$O(1)$

4. Decide which man (m or m') is preferred by <u>w</u>.

$$WomanPref_i = \boxed{3 \mid 8 \mid 4 \mid 22 \mid 1 \mid} \cdots$$

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

WomanRanking = | 5 | | 1 | 3 | | |

$O(n^2)$ + $O(n^2)$ = $O(n^2)$

preprations to create ranking array

G-S iterations

overall complexity