Sources of Information

Piazza: Syllabus
Contact info
Office hours
Exam dates
Lecture Notes
Links to lecture videos
HW assignments

Discussion Board
Any other reference material

DEN (D2L): HW submissions
Exams

Roles & Responsibilities

- Instructor — lectures
- TAs
- Course producers
- CS Dept Advisors
- DEN Support

Textbooks

- Algorithm Design by Jon Kleinberg & Eva Tardos

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

Roles and Responsibilities

- Instructor
- TAs
- Course producers
- Graders
- CS Dept. Advisors

Textbooks

- Algorithm Design by Jon Kleinberg & Eva Tardos

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

Grading Scale

90 - 100	A	60 - 64	C^+
86 - 89.999	A^-	55 - 59	C
80 - 85	B^+	50 - 54	C^-
70 - 79	B	45 - 49	D
65 - 69	B^-	Below 45	F

Scale will be adjusted if median falls below 75.

Exam 1	45%	July 23
Exam 2	55%	Aug. 11
	100%	

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

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

Exam1

- Network Flow
- Computational Complexity theory
- ? Approximation Methods
- Linear programming

Exam2

Corrections

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

Kharazmi 780-850

↓

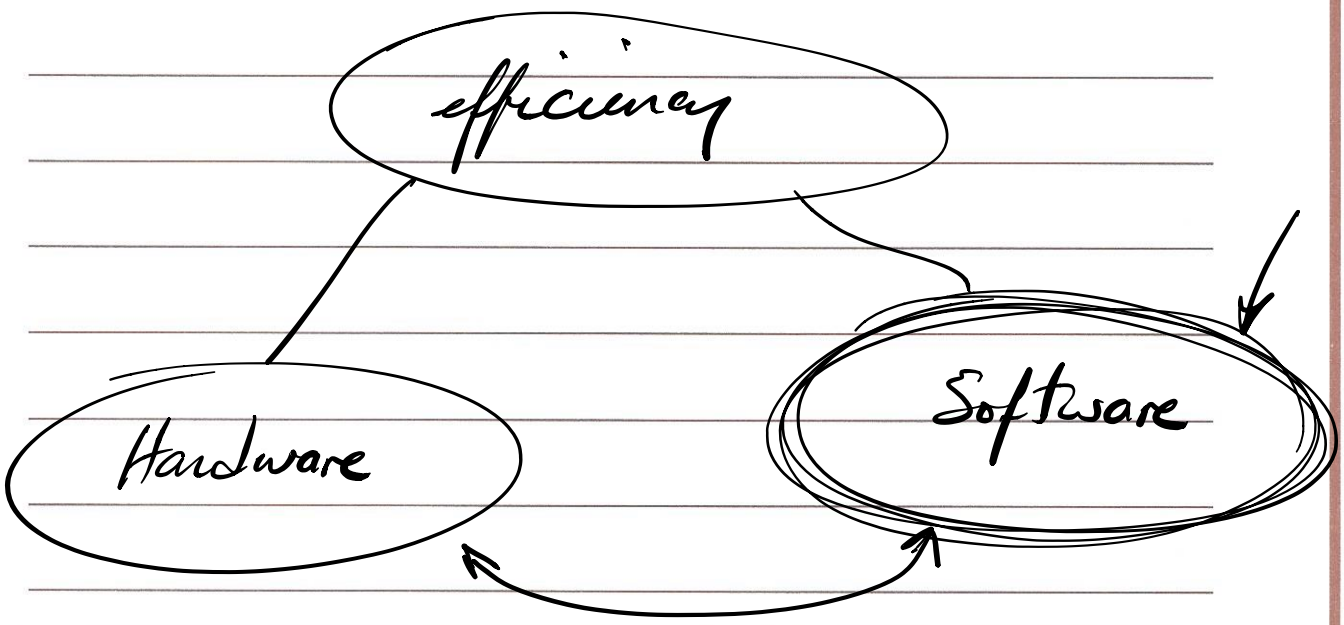
Algorithm

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

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

Correctness

efficiency



parallel processing
DMP
SMP
...
memory hierarchy
GPU
⋮

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

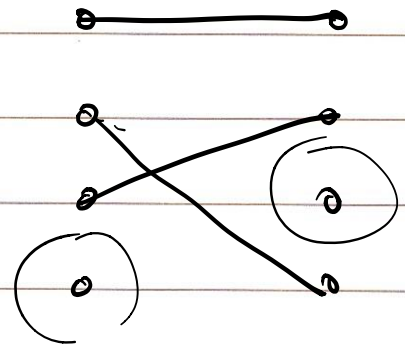
Problem: We are interested in matching $\underline{n}$ men with $\underline{n}$ women so that they could stay happily married ever after.

Step 1: Come up with a concise problem statement.

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

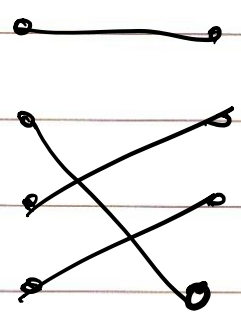
We have a set of $\underline{n}$ women, $W = \{w_1, \ldots w_n\}$

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



← A matching

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

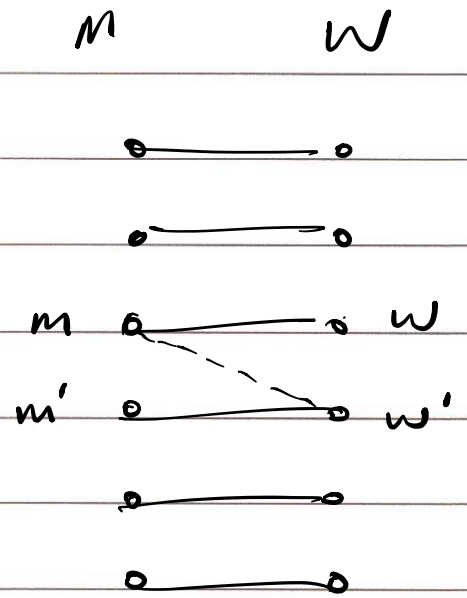
- $\underline{m}$ prefers $\underline{w}$ to $\underline{w'}$ if $\underline{m}$ ranks $\underline{w}$ higher than w'.

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

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

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

S



<u>Def.</u> Such a pair (m, w') is an <u>instability</u> 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, ~~can~~ and can be rejected repeatedly only to settle for a lower ranking woman.

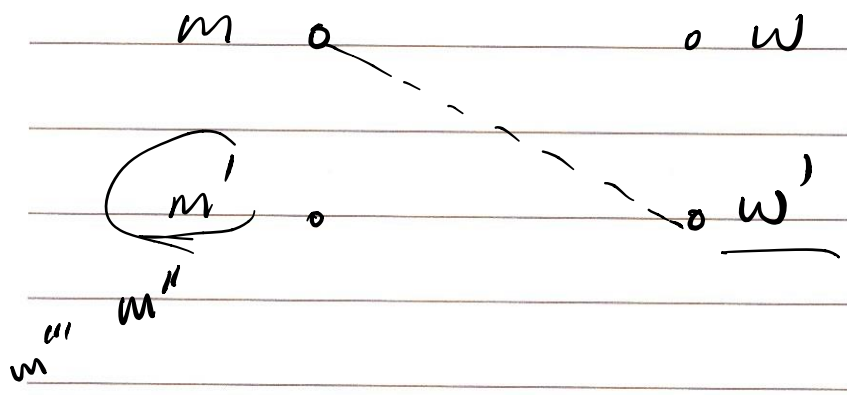
③ The algorithm terminates in at most n^2 iterations.

④ The result is a perfect matching

⑤ " " " " stable "

Proof by Contradiction

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

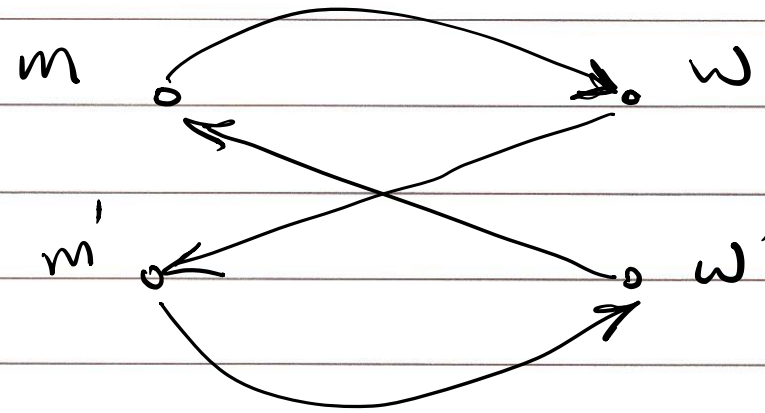


Q: Did $\underline{m}$ propose to w' at some point in the execution?

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!



if men propose $\longrightarrow (m,w),(m',w')$

if women " $\longrightarrow (m,w'),(m',w)$