Fri Sep 14 class is moved to 103 Churchill.
(only one class)

# CS3000: Algorithms & Data
# Jonathan Ullman

Lecture 2:

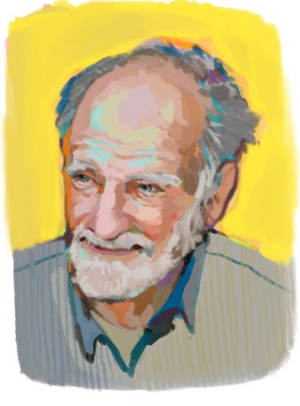
- Stable Matching: the Gale-Shapley Algorithm

Sep 11, 2018

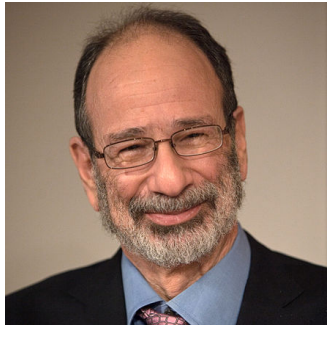
# National Residency Matching Program

- National system for matching US medical school graduates to medical residencies
  - Roughly 40,000 doctors per year
  - Assignment is almost entirely algorithmic

David Gale (1921-2008)
PROFESSOR, UC BERKELEY



Lloyd Shapley
PROFESSOR EMERITUS, UCLA



Alvin Roth
PROFESSOR, STANFORD

# Labor Markets

- Most labor markets are frustrating
  - Not everyone can get their favorite job
  - The market is decentralized

- Decentralized labor markets are confusing

You get an offer from your 2nd choice employe

Whatever you do can be wrong.

Mistakes can have ripple effects.

# Centralized Labor Markets

- What if we could just assign jobs?
  - What information would we want?

    Every doctor's preferences (how many?)

    Every hospital's preferences

    How many doctors / hospitals

    ranking or ordinal preferences
  - How would we choose the assignment?

    Maximize overall happiness?

    Stability

# Matchings

- We are given the following information
  - $n$ doctors $d_1 \dots d_n$
  - $n$ hospitals $h_1 \dots h_n$
  - each doctor's ranking of hospitals $d_1 : h_2 \succ h_3 \succ h_1$
  - each hospital's ranking of doctors $h_1 : d_1 \succ d_3 \succ d_2$

$d_1$'s preferences

| | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|
| MGH | Bob | Alice | Dorit | Ernie | Clara |
| BW | Dorit | Bob | Alice | Clara | Ernie |
| BID | Bob | Ernie | Clara | Dorit | Alice |
| MTA | Alice | Dorit | Clara | Bob | Ernie |
| CH | Bob | Dorit | Alice | Ernie | Clara |

| | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|
| Alice | CH | MGH | BW | MTA | BID |
| Bob | BID | BW | MTA | MGH | CH |
| Clara | BW | BID | MTA | CH | MGH |
| Dorit | MGH | CH | MTA | BID | BW |
| Ernie | MTA | BW | CH | BID | MGH |

# Matchings

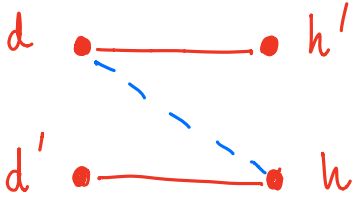
- A matching $M$ is a set of doctor-hospital pairs
  - $M = \{ (d_1, h_2), (d_2, h_3) \}$
  - matching: no doctor/hospital appears twice
  - perfect matching: every doctor/hospital appears once
  - "$d$ is matched to $h$": $(d, h) \in M$

"d is matched": $(d, h) \in M$ for some $h$

# Stable Matchings

- A matching $M$ is unstable if some doctor-hospital pair prefer one another to their mate in $M$

- Instabilities
  1. $d, h$ such that $d$ is matched to $h'$, $h$ is unmatched, but $d : h \succ h'$
  2. $d, h$ such that $h$ is matched to $d'$, $d$ is unmatched, but $h : d \succ d'$
  3. $d, h$ such that $d$ is matched to $h'$, $h$ is matched to $d'$, but $d : h \succ h'$ and $h : d \succ d'$




d: h > h'

h: d > d'

# Ask the Audience

$M = \{ (\text{Alice}, \text{BID}), (\text{Bob}, \text{MGH}), (\text{Clara}, \text{BW}) \}$

- Either find a stable matching or convince yourself that there is no stable matching

| | 1st | 2nd | 3rd |
|---|---|---|---|
| MGH | Alice | Bob | Clara |
| BW | Bob | Clara | Alice |
| BID | Alice | Clara | Bob |

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Alice | BW | BID | MGH |
| Bob | BID | MGH | ~~BW~~ |
| Clara | MGH | BID | BW |

$M' = \{ (\text{Bob}, \text{MGH}), (\text{Alice}, \text{BW}), (\text{Clara}, \text{BID}) \}$

$M'' = \{ (\text{Alice}, \text{BW}), (\text{Bob}, \text{BID}), (\text{Clara}, \text{MGH}) \}$

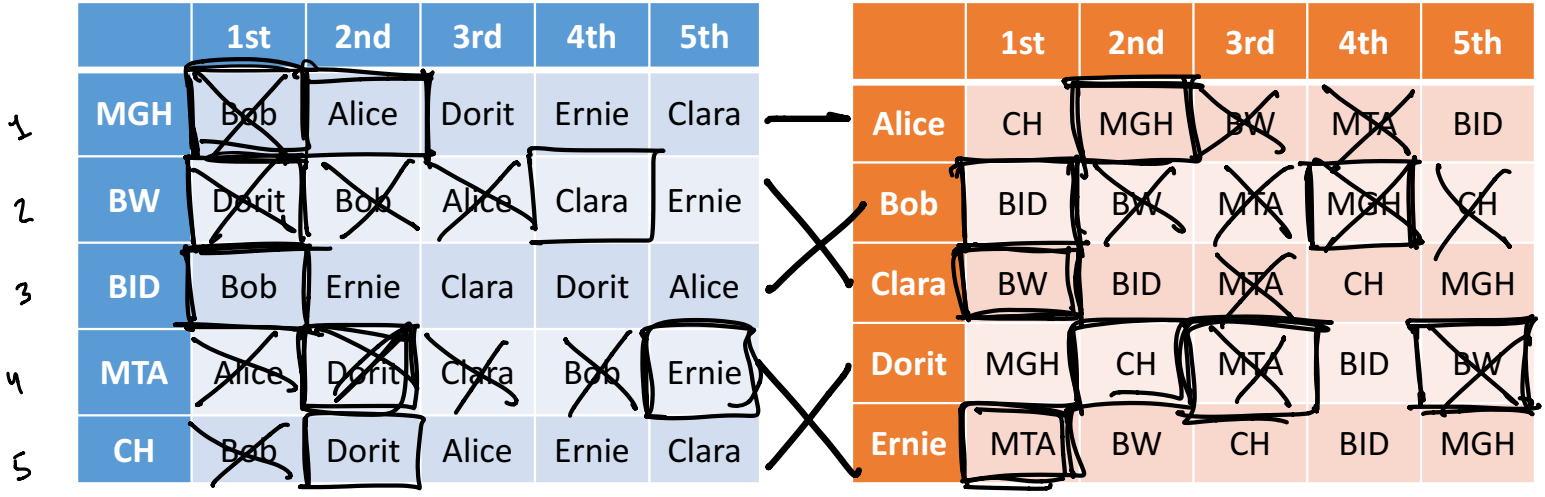
# Gale-Shapley Algorithm (Deferred Acceptance)

Initially any hospital

- Let M be empty
- While (some hospital h is unmatched):
  - If (h has offered a job to everyone): break
  - Else: let d be the highest-ranked doctor to which h has not yet offered a job
  - h makes an offer to d:
    - If (d is unmatched):
      - d accepts, add (d,h) to M
    - ElseIf (d is matched to h′ & d: h′ > h):
      - d rejects, do nothing
    - ElseIf (d is matched to h′ & d: h > h′):
      - d accepts, remove (d,h′) from M and add (d,h) to M
- Output M

A "job offer"

# Gale-Shapley Demo



# Observations

- Hospitals make offers in descending order

  If $(h, d) \in M$ at some point. $h$ will never be matched to a doctor it likes better.

- Doctors that get a job never become unemployed

  If $d$ is matched at some point, then it's matched forever.

- Doctors accept offers in ascending order

  If $(d, h) \in M$ at some point. $d$ ends up with a hospital it prefers to $h$.

# Gale-Shapley Algorithm

- Questions about the Gale-Shapley Algorithm:
  - Will this algorithm terminate?
  - Does it output a perfect matching?
  - Does it output a stable matching? (Do stable matchings exist?)
  - How do we implement this algorithm efficiently?

# GS Algorithm: Termination

- **Claim:** The GS algorithm terminates after $n^2$ iterations of the main loop ($n^2$ job offers)

There are only $n^2$ possible job offers.

No offer is made twice.

# GS Algorithm: Perfect Matching

- **Claim:** The GS algorithm returns a perfect matching (all doctors/hospitals are matched)

Proof by Contradiction:

- Suppose some $h$ is unmatched
- $\Rightarrow$ some $d$ is unmatched
- case ①: $h$ never offered to $d$
  $\Rightarrow$ algorithm wouldn't end

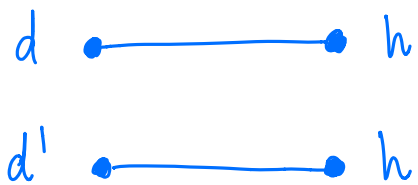
case ② $h$ made an offer to $d$

- case ②a $d$ accepted $\Rightarrow$ but then $d$ would be matched
  case ②b $d$ rejected $\Rightarrow$ but then $d$ would be matched □

# GS Algorithm: Stable Matching

stable matchings always exist?

- **Stability:** GS algorithm outputs a stable matching
- Proof by contradiction:
  - Suppose there is an instability $d, d', h, h'$




d: $h > h'$

h: $d > d'$

We know h made a job offer to d before d'

case ①: d accepted at the time

case ②: d rejected at the time

Contradiction will be that in either case d: $h' > h$

# GS Algorithm: Stable Matching

stable matchings always exist?

- **Stability:** GS algorithm outputs a stable matching
- Proof by contradiction:
  - Suppose there is an instability $d, d', h, h'$


case ①: d accepted at the time

B/c doctors "go up" d: h' > h

# GS Algorithm: Stable Matching

stable matchings always exist!

- **Stability:** GS algorithm outputs a stable matching
- Proof by contradiction:
  - Suppose there is an instability $d, d', h, h'$


case (2): d rejected at the time

- d rejected b/c d was matched to some $h''$

and $d: h'' > h$ ... b/c "doctors go up" $d: h' > h''$

therefore $d: h' > h$

# GS Algorithm: Running Time

- **Running Time:**
  - A straightforward implementation requires $\approx n^3$ operations, $\approx n^2$ space

# GS Algorithm: Running Time

- Let M be empty
- While (some hospital h is unmatched):
  - If (h has offered a job to everyone): break
  - Else: let d be the highest-ranked doctor to which h has not yet offered a job  1 operation
  - h makes an offer to d:
    - If (d is unmatched):
      - d accepts, add (d,h) to M
    - ElseIf (d is matched to h' & d: h' > h):
      - d rejects, do nothing
    - ElseIf (d is matched to h' & d: h > h'):
      - d accepts, remove (d,h') from M and add (d,h) to M
- Output M

Loop runs at most $n^2$ times

$\leq n$ ops to compare $h, h'$

$(n^2$ job offers$) \times (n$ per offer$) = n^3$ time

# GS Algorithm: Running Time

- **Running Time:**
  - A careful implementation requires just $\approx n^2$ time and $\approx n^2$ space

# GS Algorithm: Running Time

- **Running Time:**
  - A careful implementation requires just $\approx n^2$ time and $\approx n^2$ space

create a an array of doctor × hospital in $n^2$ ops

| | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|
| Alice | CH | MGH | BW | MTA | BID |
| Bob | BID | BW | MTA | MGH | CH |
| Clara | BW | BID | MTA | CH | MGH |
| Dorit | MGH | CH | MTA | BID | BW |
| Ernie | MTA | BW | CH | BID | MGH |

| | MGH | BW | BID | MTA | CH |
|---|---|---|---|---|---|
| Alice | 2nd | 3rd | 5th | 4th | 1st |
| Bob | 4th | 2nd | 1st | 3rd | 5th |
| Clara | 5th | 1st | 2nd | 3rd | 4th |
| Dorit | 1st | 5th | 4th | 3rd | 2nd |
| Ernie | 5th | 2nd | 4th | 1st | 3rd |

# GS Algorithm: Running Time

- **Running Time:**
  - A careful implementation requires just $\approx n^2$ time and $\approx n^2$ space

    - $n^2$ time to convert (doctor) × (rank) ⇒ (doctor)[rank]
    - ($n^2$ offers) × (2 operations)

    ---

    $\propto n^2$ operations

# Real World Impact

TABLE I

STABLE AND UNSTABLE (CENTRALIZED) MECHANISMS

| Market | Stable | Still in use (halted unraveling) |
|---|---|---|
| American medical markets | | |
| NRMP | yes | yes (new design in '98) |
| Medical Specialties | yes | yes (about 30 markets) |
| British Regional Medical Markets | | |
| Edinburgh ('69) | yes | yes |
| Cardiff | yes | yes |
| Birmingham | no | no |
| Edinburgh ('67) | no | no |
| Newcastle | no | no |
| Sheffield | no | no |
| Cambridge | no | yes |
| London Hospital | no | yes |
| Other healthcare markets | | |
| Dental Residencies | yes | yes |
| Osteopaths (<'94) | no | no |
| Osteopaths (≥'94) | yes | yes |
| Pharmacists | yes | yes |
| Other markets and matching processes | | |
| Canadian Lawyers | yes | yes (except in British Columbia since 1996) |
| Sororities | yes (at equilibrium) | yes |

Table 1. Reproduced from Roth (2002, Table 1).

# Real World Impact

- **Doctors ↔ Hospitals**
  - Have to deal with two-body problems
  - Have to make sure doctors do not game the system
- **Kidneys ↔ Patients**
  - Not all matches are feasible (blood types)
  - Certain pairs must be matched
- **Students ↔ Public Schools**
  - Siblings, walking zones, diversity
- **Reform Rabbis ↔ Synagogues**
  - No idea, just a fun example

# Ask the Audience

- **Claim:** For every $n \in \mathbb{N}$, $\sum_{i=0}^{n-1} 2^i = 2^n - 1$

- **Proof by Induction:**