- HW0, HW1 released.

- Recitations begin the week of Feb 1.

- If you have trouble accessing Canvas, Gradescope, OTHQ, please make a private post on Piazza.

- Enroll yourself on Piazza.

- Violation of course policies will be dealt with harshly.

- OH today 1-2 pm. (link on Piazza)

Stable Matching:

**Input:**
- $X$: set of $n$ vertices
- $Y$: set of $m$ vertices

Each vertex in $X$ has a preferences list ranking all vertices in $Y$. 
Each vertex in Y has a preference list ranking all vertices in X.

**Stable** Output: A matching in which each vertex in X is paired with exactly one vertex in Y.

\[
\begin{align*}
X & \quad Y \\
(8, 4, 1, 15, \ldots) & \quad (x, h, i, \ldots) \\
\end{align*}
\]

Permute vertices in Y. \( \pi \): permutation.

\[
\begin{align*}
a - \pi(1), & \quad b - \pi(2), \quad c - \pi(3), \ldots
\end{align*}
\]
Stable matching: matching that does not have an instability.

Q1: Does every instance have a stable matching?

Q2: Can an instance have exactly one stable matching & can an instance have >1 stable matches?

Q3: If an input has a stable matching, can we find one?
Ans to Q.2

\[(y, y') \quad \xrightarrow{\text{x}} \quad \begin{array}{c} y \quad \xrightarrow{\text{x'}} \quad (x, x') \end{array} \]

\[(y, y') \quad \xrightarrow{\text{x'}} \quad \begin{array}{c} y' \quad \xrightarrow{\text{x}} \quad (x', x) \end{array} \]

\[(y', y) \quad \xrightarrow{\text{x'}} \quad \begin{array}{c} y' \quad \xrightarrow{\text{x}} \quad (x', x) \end{array} \]

\[
\begin{array}{l}
a \quad \xrightarrow{\text{1}} \\
b \quad \xrightarrow{\text{2}} \\
c \quad \xrightarrow{\text{3}} \\
d \quad \xrightarrow{\text{4}} \\
e \quad \xrightarrow{\text{5}} \\
\end{array}
\]

3 decide whether to be with c or with e.
GS (Gale-Shapley)

- Initially all vertices are free.

- While there is a free vertex $x \in X$ s.t.
  - $x$ has not yet proposed to all vertices in $Y$ do

  $y \leftarrow$ highest ranked vertex in $Y$ whom
  - $x$ has not yet proposed to

  if $y$ is free then

  $(x, y)$ becomes a pair

  else if $(x', y)$ exists then

  if $y$ prefers $x'$ to $x$ then
(x, y) becomes a pair
x' becomes free

- return all pairs

Q. Will the alg. always terminate?

Yes

- each vertex in X makes \( \leq 1 \) proposal to a vertex \( y \in Y \).

- Thus \( \leq n \) proposals for \( n \)

- \( |X| = n \)

- \( \leq n^2 \) proposals, i.e., GS alg. ends after \( \leq n^2 \) iterations of the while loop.

Observation: Once a vertex \( y \in Y \) receives
their first proposal \( y \) will never be free. As the algorithm progresses, \( y \)'s partner can only get better.

At any point in the alg

**Observation 2**: Every vertex in \( X \cup Y \) is paired with at most one vertex.

**Lemma**: GS alg. outputs a perfect matching.

**Proof**: Assume \( \Diamond \). Let \( x \in X \) be a free vertex at the end of the GS alg. This means that \( x \) has proposal to all vertices in \( Y \). By Obs 1, all vertices in \( Y \) are
paired up. Thus \( \leq n-1 \) vertices in \( X \) are paired with \( m \) vertices in \( Y \). Hence, by \( \text{FIP} \) there must be a vertex on \( X \) paired up with two vertices in \( Y \), contradicting Obs 2.

**Case 2:** \( y \in Y \) is free.

Thus: G-S alg. outputs a stable matching.

**Proof:** Assume otherwise.

\((\ldots, y', \ldots, y, \ldots)\)

- \( x \) must have proposed to \( y' \) before it
propose to \( y \).

- Since \( x \) ends up with \( y \), it must be that \( x \) got rejected by \( y' \).

- \( y' \) rejects \( x \) because \( y \) \( x'' \). Thus \( y' \) prefers \( x'' \) over \( x \).

- \underline{Case I}: \( x' = x'' \)
  
  Contradicts that \( x \) is ranked higher than \( x' \).

- \underline{Case II}: \( x' \neq x'' \)

  By Obs 1, \( x' \) is ranked higher than \( x'' \) on \( y' \)'s list & hence higher than \( x \), contradicting that \( x \) is ranked higher than \( x' \).
valid(x) = \{ y ∈ Y | \exists a stable matching containing (x, y) as a pair \}

Best(x) = \arg\max_y \{ y | (x, y) \text{ is valid} \}

- y ∈ valid(x)
- ∃ y’ ∈ Y s.t. y’ is ranked higher than y on x’s list, y’ ∉ valid(x)

S* = \{(x, Best(x)) | x ∈ X\}

**Theorem:** Every execution of the GS alg output S*.

**Proof:** Assume otherwise. Let E be an execution of the alg. In which some vertices in X get rejected
by their best valid partners. Among all such vertices, let \( x \in V \) have the honor of being the first vertex to be rejected by \( \text{Best}(x) \).

Let \( y = \text{Best}(x) \).

**Why did \( y \) reject \( x \)?**

*Because \( f \neq x' \).*

By defn. \( f \) valid(.) there exits a stable matching \( S \) in which \( (x, y) \) is a pair.

\((..., x', ..., x, ...)\)
\text{Valid}(y) = \{ x \in X \mid (x, y) \text{ is a pair in some stable matching} \}
\text{worst}(y) \triangleq x \iff y \\
\ x \in \text{valid}(y) \\
\exists x' \in X \text{ st. } x' \text{ is ranked below } x \text{ in the prefix list of } y, \ x' \notin \text{valid}(y).

Thus: G-S alg. always pairs winning with worst$(y)$. 