Announcements

- H.W 1 is out on CMS. Due next Thursday, 11:59 pm.
    - 1 coding problem: implementing the Gale-Shapley algorithm
    - 2 theoretical problems: Supply proofs, algorithm, ...

      Amount of details:
        - Provide complete mathematical proofs.
        - Describe your algorithm / pseudocode in English. Supply proof of correctnes, run time analysis.
- Start early!
- Use Piazza, TA office hours, Course Website, Book.
- Please Type set your HW submissions: Latex or Word. → CMS
- Collaboration policy: Can form groups up to 4 to solve HWs. In fact it is encouraged and is a great way to learn!
    - For people who are having difficulty finding partners, STAY TUNED.
    - Acknowledge collaborators in HW submission.
- Academic Integrity: (i) Write your own solutions. Do not plagiarize.

  (ii) Web sources: - DO NOT use for searching solutions or code.
    - Accidental discovery: stop reading!
    - Can use for learning & supplementing material taught in class. Cite any resource you used from web in H.W. Will not get credit for that part.

  We are aware of HW cheat sites.
- Slip days: (a) 6 days in total. Has to be used as whole days.

  (b) Max 3 days / HW

  Use wisely! (save them for actual emergencies & busy weeks)
- TA office Hours: stay tuned.

Stable Matching [Section 1.1]

This problem is motivated by a very real world problem: Matching job applicants with employers.

Set up: n employers $\qquad$ m job applicants

Assume each employer has a ranking of the job applicants. (No ties)

Assume each job applicant has a ranking of the employers.

→ They submit it to a central platform.

Task: Algorithm for the central platform to match employers with job applicants.

Our setup: n employers and n job applicants

Each employer requires exactly one applicant.

$$E = \{e_1, e_2, \dots, e_n\} \qquad A = \{a_1, \dots, a_n\}.$$

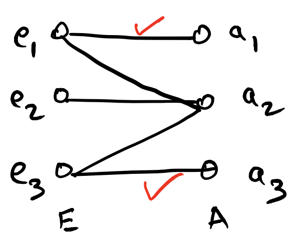
$$E \times A = \{(e_i, a_j) : e_i \in E,\ a_j \in A\}.$$

Output: A perfect matching in $E \times A$.

Matching: $M \subseteq E \times A$ in which any $e \in E$ appers at most once in M; any $a \in A$ appears at most once in M.

Perfect Matching:

$M \subseteq E \times A$ in which any $e \in E$ appers exactly once in M; any $a \in A$ appears exactly once in M.

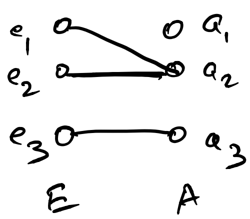


$$\{(e_1, a_1),\ (e_3, a_3)\} = M$$

$$M' = M \cup \{(e_2, a_2)\}$$

Notions of niceness.

(i) Maximize # of top candidates for employers.



Stability [Gale and Shapley, 1962]

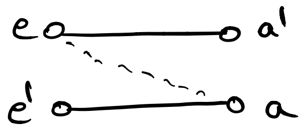
$E = \{e_1, \dots, e_n\}$ $\qquad$ $A = \{a_1, \dots, a_n\}$.

Let M be a perfect matching.

(e,a) is unstable (w.r.t M) if:

(i) $(e,a) \notin M$.

(ii) e and a prefer each other compared to their pairings in M.



$e: a > a'$

$a: e > e'$.

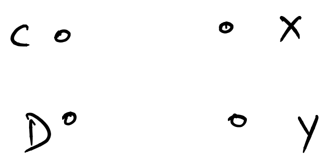
$(e \neq e', a \neq a')$.

Def^n: M is a stable matching if

(i) M is a perfect matching &

(ii) M has no unstable pairs.

Example: 1



$C: X > Y$

$D: X > Y$

$X: C > D$

$Y: C > D$

M_1: (C, Y), (D, X) unstable.

(D, X) is an unstable pair

M_2: (C, X), (D, Y)

② $C \circ \qquad \circ X$

$D \circ \qquad \circ Y$

$C: X > Y$

$D: Y > X$

$X: D > C$

$Y: C > D$

M_1: (C, Y), (D, X) stable.

M_2: (C, X), (D, Y) stable.