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

Study problem steps:

1. Come-up with a concise problem statement
2. Present a solution
3. Prove correctness
4. Perform complexity analysis

Stable matching. 稳定匹配.

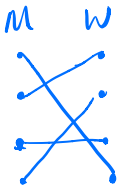
problem: match n men & n woman

step 1: come up with a concise problem statement

n men, $M=\{m_1, \cdots, m_n\}$

n woman, $M=\{w_1, \cdots, w_n\}$

perfect matching:



matching with property that each member of M and each member of W appear in exactly one pair in S'

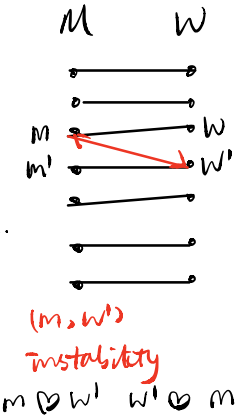
notation of preference:

Each man $m \in M$ ranks all women :

- m ♡ w to w': if m rank w higher than w'
- ordered ranking of m is his preference

  $pm_i = [w_{i1}, w_{i2} \cdots w_{in}]$

same for women, each woman $w \in W$ ranks all men ...



matching S is stable if ① it is perfect

② no instabilities

step 2

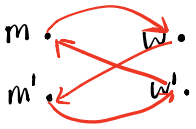
Gale-Shapley:

GALE–SHAPLEY (*preference lists for men and women*)

INITIALIZE S to empty matching.
WHILE (some man m is unmatched and hasn't proposed to every woman)
  $w \leftarrow$ first woman on m's list to whom m has not yet proposed.
  IF (w is unmatched)
    Add pair m–w to matching S.
  ELSE IF (w prefers m to her current partner m')
    Remove pair m'–w from matching S.
    Add pair m–w to matching S.
  ELSE
    w rejects m.

RETURN stable matching S.

stable matching



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

women proposing (m, w')(m', w)

step 3: proof of correctness

① woman's perspective: starts single, once she get engaged, and she can only get into better engagement wrt her preference list.

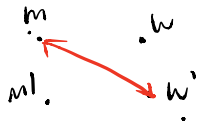
② man's perspective: starts single, once he get engaged, and may get dropped repeatedly only settle for a lower ranking woman

③ solution will terminate in at most n^2 interactions

④ solution is perfect matching stable ...

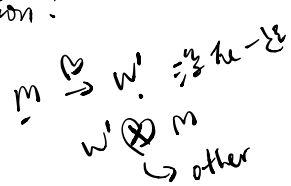
proof by contradiction:

Assume instability exist.



Did m propose to w' at some point in execution?

- if no. m prefer w → contradiction
- if yes, w' reject m → contradiction



Step 4 Complexity analysis.

① Identify a free man. O(1)

② Identify the highest ranked woman to whom m has not yet proposed. O(1)

	get	post
1. Array	O(1)	O(1)
2. linked list	O(1)	O(1)
3. stack	O(1)	O(1)
4. queue	O(1)	O(1)

Keep an array Next[1, ..., n] where Next[m] points to the position of the next women, he will be proposing to on his preference list.

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

ManPref[m, i] denotes the ith woman on man m's preference list.

m ♡ [... i ...]
list w

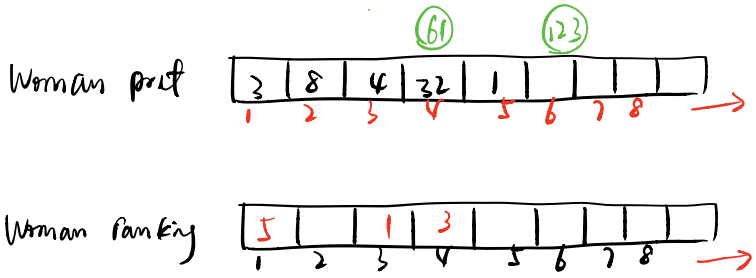
To find next women w to whom m will be proposing to:

w = ManPref[m, Next[m]] → takes O(1)

③ Determine Woman w's status O(1)

Keep an array called current[1 ... n] current[w] is Null if w is single & set to m if w is engaged to m (Taken O(1))

④ Determine which man is preferred by w?



preparation before entering GS iteration

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

preparation + GS iterations overall complexity $= O(n^2)$

$O(n^2)$ $O(n^2)$

Def: Woman w is a valid partner of a man m if there is a stable matching that contains the pair (m, w)

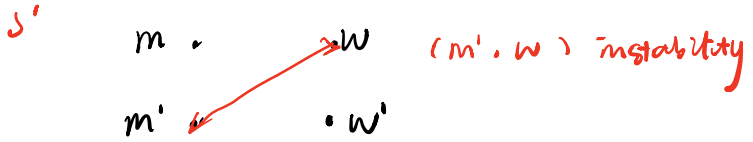
Def: m's best valid partner is the valid partner with the highest ranking in m's preference list.

Claim: Every execution of GS (when men propose) results in the same stable matching regardless of order in which man propose

Plan: prove this. show that when men proposed, they always end up with the best valid partner.

proof by contradiction:

say m is 1st man rejected by a valid partner w (in favor of m')



claim: When men propose, women end up with their worst valid partners.

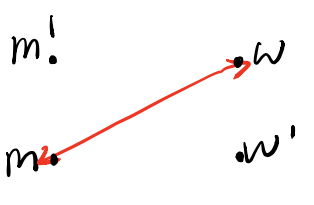
Proof: By contradiction ...

suppose we end up with a matching S

where for pair (m, w) in S, m is <u>not</u> w's worst valid partner

so there must be another matching S'

w is paired with a man m' she likes less.



(m, w) is instability