Feb 19

Implementing GS

$\boxed{\text{Initialization}} \leftarrow T_0$

While $(\ldots) \leftarrow \#\text{iters} = T_1 \leq n^2$

$\boxed{\text{Body}} \leftarrow T_2$ (for each iter.)

$\boxed{\text{Output } S} \leftarrow T_3$

Overall runtime $\leq T_0 + T_1 \cdot T_2 + T_3 \Longrightarrow$ if we could assum

$\leq O(n^2) + n^2 \cdot O(1) + O(n)$ $\quad T_1$ is $O(n^2)$

$= O(n^2) + O(n^2) + O(n) = O(n^2)$ $\quad T_2$ is $O(1)$

T_3 is $O(n)$

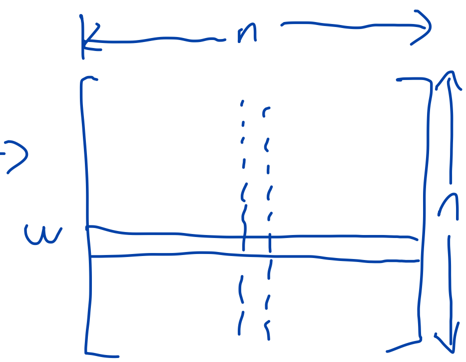
Notation change: Assume $M = [n] \stackrel{def}{=} \{1, 2, \ldots n\}$

$\{m_1, m_2, \ldots m_n\} \mapsto \{1, 2 \ldots n\}$ $W = [n]$

→ Array indices start at 1

Q0) How is the input represented?

2D-Array ManPref WomanPref →



WomanPref[w][i]

ManPref[m][i] = ID of the i-th prefferred woman for m

WomanPref[w][i] = ID of the i-th preferred men for w

Initialization: n/a

Query: Read value at a specific location WomanPref[w][i] → $O(1)$

Update: n/a

Q1) How do we find a free woman w?

A1) Maintain a linked list of free women, call <u>free</u>

Init: Add all women to <u>free</u> ← $O(n)$

Query: Pick 1st woman in <u>free</u> (+ delete the entry) ← $O(1)$

Update: Case 1: m was free → do nothing

Case 2.1: (m, w') remain engaged → Add w to <u>free</u>

Case 2.2: (m, w) get engaged → Add w' to <u>free</u>

$O(1)$