Feb 19

# Implementing GS

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

While ( ... ) $\leftarrow$ #iters $= T_1 \le n^2$

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

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

Overall runtime $\le T_0 + T_1 \cdot T_2 + T_3$ $\Rightarrow$ if we could assume $T_1$ is $O(n^2)$, $T_2$ is $O(1)$, $T_3$ is $O(n)$

$\le O(n^2) + n^2 \cdot O(1) + O(n)$

$= O(n^2) + O(n^2) + O(n) = O(n^2)$

---

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

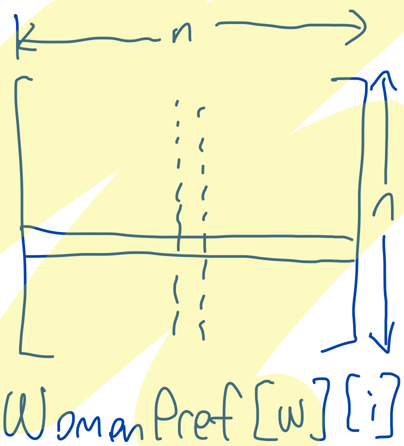
$\{m_1, m_2, \dots m_n\} \mapsto \{1, 2 \dots n\}$ $W = [n]$

$\rightarrow$ Array indices start at 1

---

Q0) How is the input represented?

2D-Array ManPref WomanPref $\rightarrow$




WomanPref[w][i]

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

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

---

Initialization: n/a

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

Update: n/a

Q1) How do we find a free woman w?

A1) Maintain a linked list of free women, call free

Init: Add all women to free ← $O(n)$

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

Update: Case 1: m was free → do nothing

Case 2.1: (m, w') remain engaged → Add w to free

Case 2.2: (m, w) get engaged → Add w' to free

$\}$ $O(1)$

Feb 22

Q2) How do we figure out w's best unproposed man m?

A2) Maintain an array Next of size n

Next[w] = rank of the man w should propose to next

Init: Next[w] = 1 $\forall$ w ← $O(n)$

Query: Who should w propose next? WomenPref[w][Next[w]] ← $O(1)$

Update: Next[w]++ ← $O(1)$

Q3) How do we figure out who m is engaged to?

A3) Array Current of length n

$$\text{Current}[m] = \begin{cases} -1 & \text{if } m \text{ is free} \\ w & \text{if } (m,w) \text{ are engaged} \end{cases}$$

Init: $\text{Current}(m) = -1 \ \forall m \leftarrow O(n)$

Query: Read $\text{Current}[m] \leftarrow O(1)$

Update: If $(m,w)$ gets engaged $\Rightarrow \text{Current}[m] = w \leftarrow O(1)$

---

Q4) If $w' > w$ in $L_m$?

Scan ManPref[m] & figure out the location of $w$ & $w'$

$\rightarrow O(n)$

$\Rightarrow$ overall GS is $O(n^3)$

$\rightarrow$ Can we make it more efficient?

$G = (V, E)$ $\qquad E \subseteq V \times V$

$V$: set of vertices/nodes

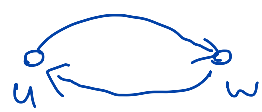
$E$: set of edges

<u>Default:</u> $n = |V|$ ; $m = |E|$

<u>Def:</u> $G$ is undirected $\iff$ $\forall\, u, w \in V$, $(u,w) \in E$ $\iff$ $(w,u) \in E$

s.t. $u \neq w$


$\equiv$




o.w. $G$ is directed