

Feb 14

Implementing GS

Initialization $\leftarrow T_0$

while (...) \leftarrow #iters = $T_1 \cdot n^2$

Body $\leftarrow T_2$ (for each iter)

Output S $\leftarrow T_3$

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

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

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

\rightarrow Array indices start at 1

Q0) How is the input represented?

2D-Array ManPref, WomanPref \rightarrow

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

WomanPref[w][i] = ID of the i-th most 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 woman, call free

Init: Add all woman to free $\leftarrow O(n)$

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

Update: Case 1: m was free \rightarrow do nothing

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

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

} $O(1)$