

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

}  $O(1)$

Feb 17  
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:  $Next[w] = 1 \quad \forall w \leftarrow O(n)$

Query: Who should  $w$  propose to next?  $WomanPref[w][Next[w]]$

$\rightarrow O(1)$

Update:  $Next[w]++ \rightarrow O(1)$

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:  $\text{Current}[m] = -1 \quad \forall m \rightarrow O(n)$

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

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

---

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

Scan  $\text{ManPref}[m]$  & figure out location of  $w$  &  $w'$   
 $\rightarrow O(n)$

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

$\hookrightarrow$  Can we make it more efficient?

Feb 17

$$G = (V, E)$$

↓  
set of vertices/nodes

↘  
set of edges

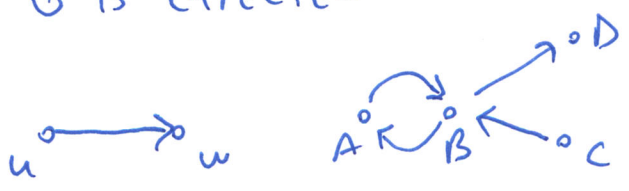
$$E \subseteq V \times V$$

Default:  $n = |V|$  ;  $m = |E|$

Def:  $G$  is undirected  $\Leftrightarrow \forall u \neq w \in V, (u, w) \in E \Leftrightarrow (w, u) \in E$



o.w.  $G$  is directed



- (.) Airline map (u)
- (.) Wikipedia page