Oct 3, 2014

TopOrd $(G = (V, E))$

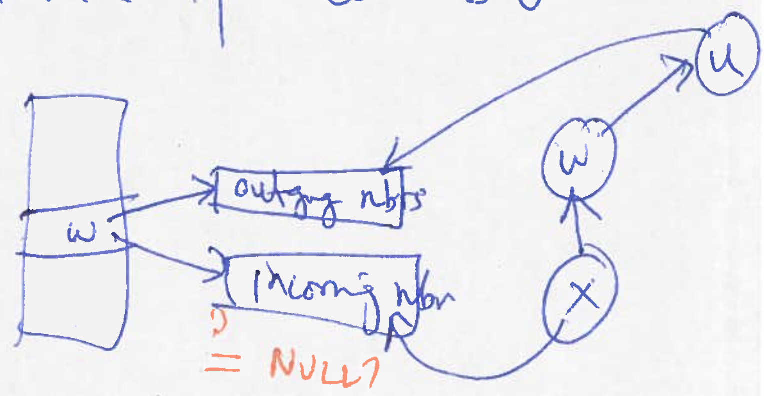
Argue: (new)

if
0. If $V = \{u\}$, output $u \leftarrow O(1)$
1. Let w be a vertex with 0 incoming edges ← $O(n)$ [check every $w \in V$]
2. $G' \leftarrow G \setminus \{w\} \leftarrow O(n)$
3. Return w; TopOrd(G')

$O(n_w)$

Q: How much time to determine if w has 0 incoming edges?

A: $O(1)$
(G is adj. list format)



Every recursive call takes $O(n)$ time.
→ How many recursive calls? $\leq n$
⇒ overall $O(n^2)$.

$O(m+n)$: 2 data structures

(i) InDeg$[w]$: # incoming edges to w in the current G.
(ii) L: linked list of all vertices with 0 incoming edges in the current G.

Initialize: (i) InDeg $O(m+n)$ [for each w count length of incoming nbr $O(n_w)$]

(ii) L: Add all w s.t. InDeg$[w] = 0 \rightarrow O(n)$

$O(m+n)$

Query: Delete front of L $O(1)$

Update: L & InDeg for G' ($G' = G \setminus \{w\}$)

(no need to delete anything else from L)

$\Rightarrow$ InDeg[u] --

$\forall u$ s.t. $(w, u) \in E$

$\Rightarrow O(n_w)$

Also check if InDeg[u] = 0 $\Rightarrow$ Add u to L

$\Rightarrow O(n_w)$

$$\text{Overall time} = \underset{O(m+n)}{\text{Init.}} + \sum_{w \in V} \underbrace{\text{Query}(w) + \text{Update}(w)}_{O(n_w)}$$

$$= O(m+n) + O\left(\sum_w (n_w + 1)\right)$$

$$= O(m+n) + O(m+n) = O(m+n)$$

$$O\left(\sum_w (n_w + 1) = \underbrace{\sum_w n_w}_{=2m} + \underbrace{\sum_w 1}_{n}\right)$$