Oct 4

Lemma 3: If $G_{=(V,E)}$ is a DAG $\Rightarrow \exists\, w \in V$ s.t. w has 0 incoming edges.

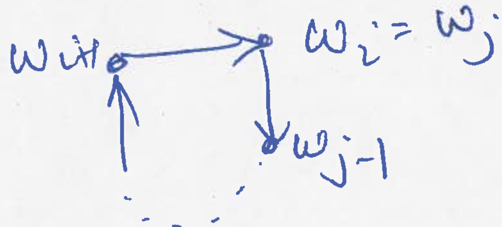
Pf (idea): By contradiction.

Assume G is a DAG but all vertices have ≥ 1 incoming edge.

$$w_{n+1} \longrightarrow w_n \;\cdots\; \longrightarrow w_3 \longrightarrow w_2 \longrightarrow w_1$$

There are $n+1$ label but there are n nodes $\underset{\text{by Pigeon hole principle (PHP)}}{\Longrightarrow}$

$\exists$ ~~i,j~~ $i < j$ s.t $w_i = w_j$



$\Rightarrow$ a cycle, contradicts G being a DAG ∎

TopOrd $(G=(V,E))$ — Correctness: Ex

$O(1)$ → 1. If $V=\{u\}$ return u

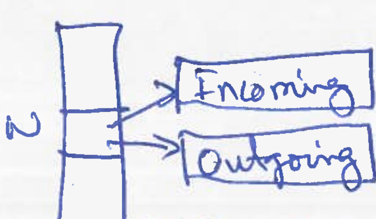
$O(n)$ 2. Let w be a vertex with 0 incoming edge

3. $G' \leftarrow G \setminus \{w\}$ ← Assume: this $O(n)$ step

4. Return w, TopOrd (G')

Runtime analysis: Try 1: $O(n^2)$, Try 2: $O(m+n)$

Q: Given $w \in V$, how quickly can you figure out if w has 0 incoming edges?



$\overset{?}{=} \phi \Rightarrow O(1)$ time : for step 2, go through all $w \in V$

$\Rightarrow O(n)$ time for each recursive call.

$\leq n$ recursive calls (since ~~each~~ each throws w in step 3)

$\Rightarrow$ Overall $\leq n \cdot O(n) = O(n^2)$ ∎

Next: $O(m+n)$

Notation: In_w & Out_w are number of incoming & outgoing neighbours of w resp. In_w: In-degree, Out_w: Outdegree

THM: $\sum_{w \in V} In_w = \sum_{w \in V} Out_w = m$.

Lemma 4: Each recursive call takes $O(In_w + Out_w + 1)$

to handle $In_w = Out_w$

$\Rightarrow$ overall $\leq \sum_{w \in V} O(In_w + Out_w + 1)$

$= O\left(\sum_w In_w + \sum_w Out_w + n\right)$

$= O(m+n)$

Pf (idea) of Lemma 4: Main obs: ONLY need to keep track of indegrees (& do not need to compute all of G' explicitly).

Data Structures:

(i) InDeg: array of length n : InDeg[w] = indegree of w in "current G"

(ii) $L \rightarrow$ linked list of all vertices w with $In_w = 0$

Initialization:

(i) InDeg[w] = In_w $\leftarrow O(In_w + 1)$

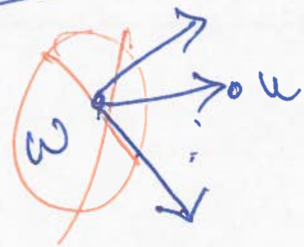
Overall: $\sum_w O(In_w + 1) = O\left(\sum_w In_w + n\right) = O(m+n)$

(ii) $L \rightarrow$ Scan through InDeg & add w to L if InDeg[w] = 0 $\Rightarrow O(n)$

$\Rightarrow$ Overall initialization: $O(m+n)$

Query: Delete the front of L to get w $\Rightarrow O(1)$ for vertex w.

Update:



Q: If we delete w what are the vertice u s.t $InDeg[u]$ need to be updated?

$\rightarrow$ for all $(w,u) \in E \leftarrow Out_w$

$$\left.\begin{array}{l} InDeg[u]-- \\ \text{If } InDeg[u] = 0 \\ \quad \text{Add } u \text{ to } L \end{array}\right\} O(1)$$

~~Overall~~ Update $= O(Out_w + 1)$.

$\Rightarrow$ Overall update / ~~que~~ query $\leq \sum_{w \in V} O(Out_w + 1)$

$= O(m+n)$ ∎