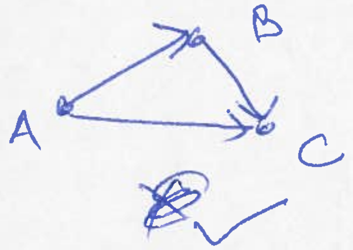
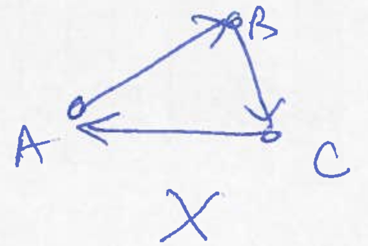
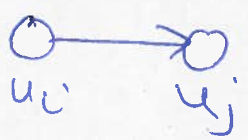
$\boxed{\text{Oct 1, 2014}}$ Directed Acyclic Graphs (DAG)

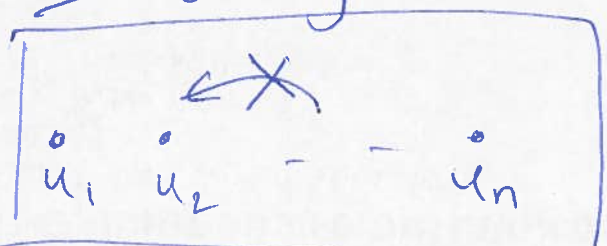
→ Directed graph $G = (V, E)$ with no (directed) cycles.

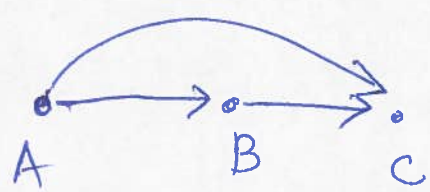




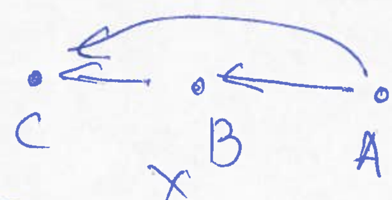
Def: Topological Ordering (Sorting) of a directed graph $G = (V, E)$ is an ordering of V
$u_1, \ldots, u_n$ s.t.
if $(u_i, u_j) \in E \Rightarrow i < j$

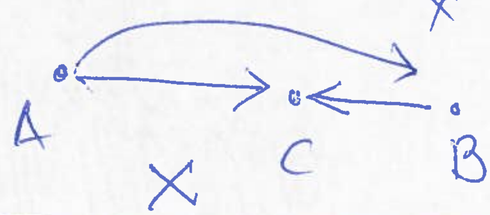






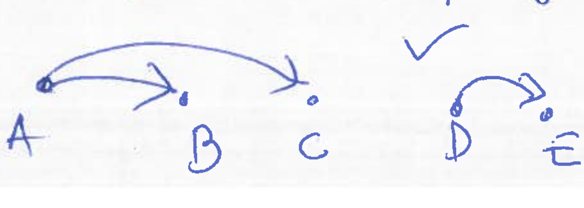
top ord ✓

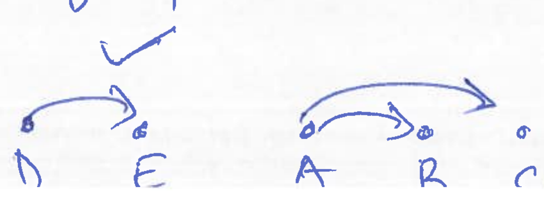




Problem: Input: Directed graph $G = (V, E)$ [DAG]

Output: A topological ordering, if one $\exists$.

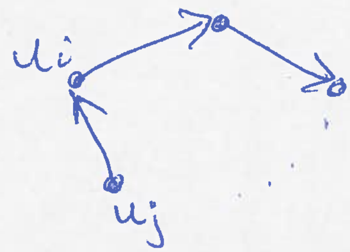




Lemma: ~~If~~ G has a topological ordering $\Rightarrow$ G is a DAG.

Pf: For the sake of contradiction, assume G has a top ordering $u_1, \ldots, u_n$ but it has a directed cycle C.

i be the smallest s.t. $u_i \in C$



~~Let~~ Since C is a cycle $\exists\, j$ s.t. $(u_j, u_i) \in E$

but by choice of i $j > i$

$\Rightarrow$ contradicts $u_1, \ldots, u_n$ being a top ordering

---

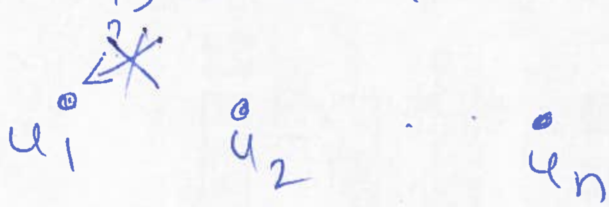
Q: If G is a DAG does it $\Rightarrow$ G has a topological ordering?

A: Yes (via an algo.)

THM: G is a DAG $\Rightarrow$ G has a topological ordering.

Idea: Reverse engineer from soln.

Assume $u_1, \ldots, u_n$ is a topological order



Q: How ~~as~~ many incoming edges into u_1?

A: 0 $\Rightarrow$ ~~idx~~ first vertex in top-ord has 0 incoming edges.

Lemma 2: If G is a DAG $\Longrightarrow$ $\exists$ a vertex w with no incoming edges.

Pf: Pf by contradiction. Assume every vertex has $\geq$ one incoming edge.

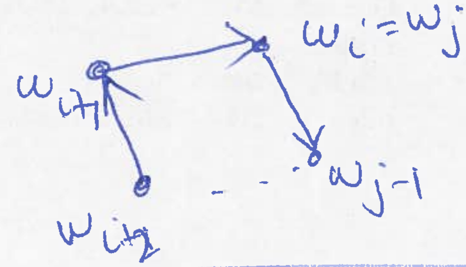
$n+1$ labels $w_1, \ldots, w_{n+1}$ but n vertices.

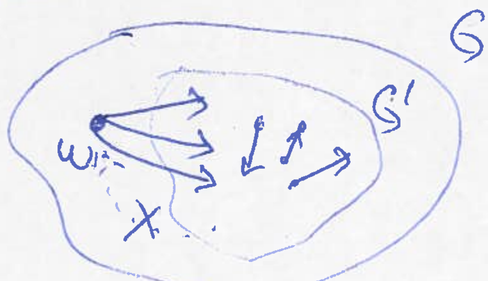
By pigeonhole principle $\exists\ i \neq j$ s.t. $w_i = w_j$

WLOG (without loss of generality) $i < j$

a cycle $\Longleftarrow$

a contradiction!





$G' = (V \setminus \{w\}, E')$

$E' = E \setminus \{(w,u) \mid u \in V\}$

Lemma 3: G' ~~has~~ is a DAG. (Pf: Ex)

$\leftarrow w'$ by applying Lemma 2 on G'

Current top order: w, w'

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

0. If $|V| = 1$ output u $(V=\{u\})$
1. Let w be a vertex with no incoming edges
2. $G' \leftarrow G \setminus \{w\}$
3. Return w; TopOrd(G')

Ex: Pf of correctness Lemmas 2 & 3