

Def A topo order for a DAG is a permutation of the nodes v_1, v_2, \dots, v_n so that for every edge (v_i, v_j) , $i < j$.

$$\underline{v_i} \rightarrow \underline{v_j}$$

Topo Order (DAG G):

L = empty list for order

While G has nodes:

let v be a node in G with no incoming edges

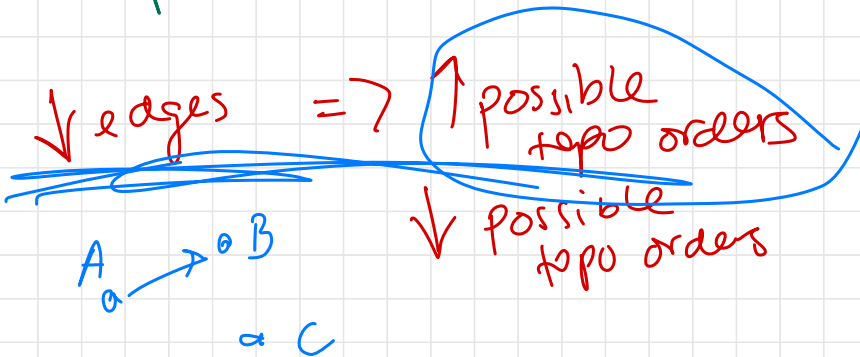
add v to L

remove v (and all its edges) from G

return L

example:

$$L = v_1, v_3, v_2, v_6, v_5, v_4$$



Proof by induction boilerplate

Claim every x has property Y .

Proof Let z be an arbitrary x . Universal Declaration

Assume that for all w smaller than z , w has property Y .

Inductive hypothesis

There are (at least 2) cases:

(Base case) Prove directly that z has property Y .

(Inductive case) Use IH to show that z has property Y .

In all cases,

z has property Y . \Leftarrow

Since z was arbitrary, all x have prop Y .

claim Topo Order finds a valid topological order on all DAGs.

proof Let G be an arbitrary DAG.

universal declaration

Assume that for all DAGs H with fewer nodes than G , Topo Order finds a valid topological order for H .

Inductive Hypothesis

There are 2 cases:

Base case: Suppose G has one node. The single node is a valid topo order for G . That node has no incoming edges and is thus returned by TopoOrder. So TopoOrder finds a valid topological order for G .

Inductive case: Suppose that G has more than one node.

By claim from quiz, there must be at least one node w/ no incoming edges. Call this node v_1 . Form G' by removing v_1 from G . Notice that G' has fewer nodes than G , so by IH, TopoOrder returns a valid topo order for G' . Call it $L' = (v_2, \dots, v_n)$. Add v_1 to front of L' to form L . We now need to check that for all edges (v_i, v_j) in G , $i < j$.

For all $i, j \neq 1$, this holds by IH. Since v_1 has no entering edges, there are no edges with $j = 1$.

So all (v_i, v_j) have $i < j$ as needed.

In all cases, TopoOrder finds
a valid topological order for G .

wrap-up
sentence