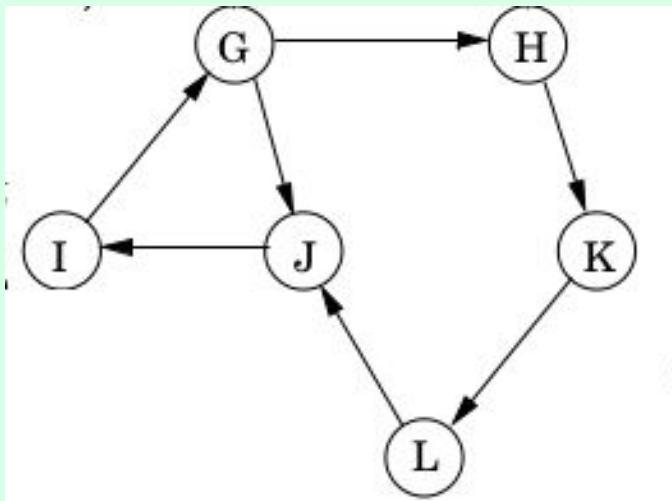




Assignment-1: **Don't delay!**



How do we find if a **directed** graph G is strongly connected?

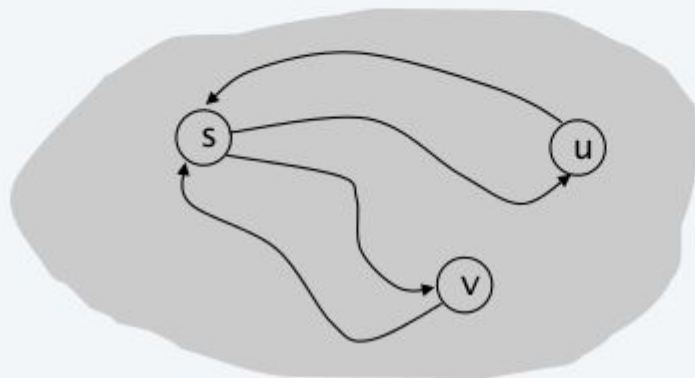


Lemma. Let s be any node. G is strongly connected iff every node is reachable from s , and s is reachable from every node.

Pf. \Rightarrow Follows from definition.

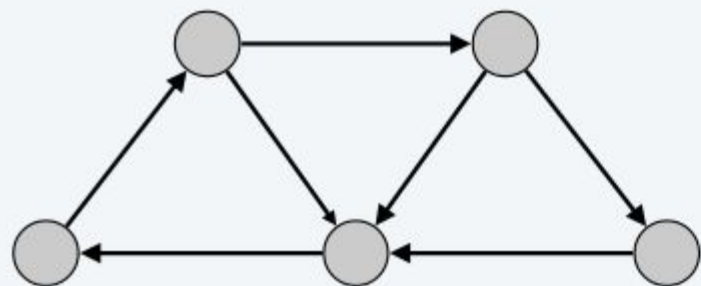
Pf. \Leftarrow Path from u to v : concatenate $u \rightarrow s$ path with $s \rightarrow v$ path.

Path from v to u : concatenate $v \rightarrow s$ path with $s \rightarrow u$ path. ■

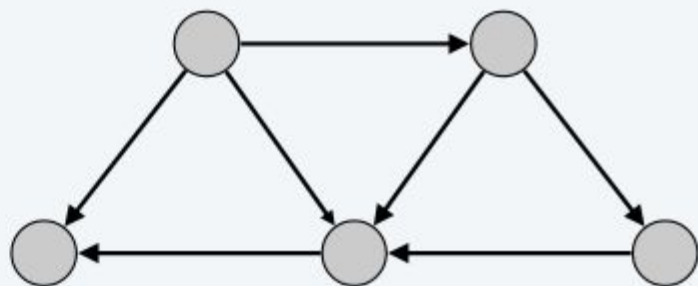


ok if paths overlap

- Pick any node s .
- Run BFS from s in G .
- Run BFS from s in $G^{reverse}$. reverse orientation of every edge in G
- Return true iff all nodes reached in both BFS executions.
- Correctness follows immediately from previous lemma. ■



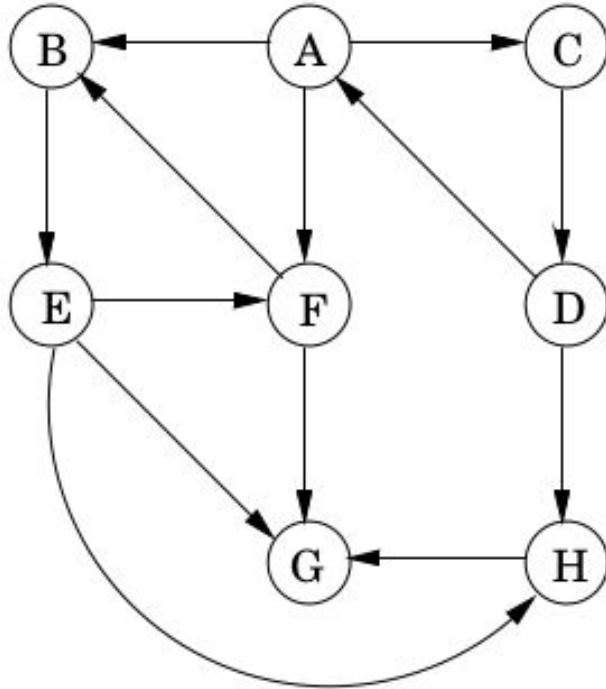
strongly connected



not strongly connected

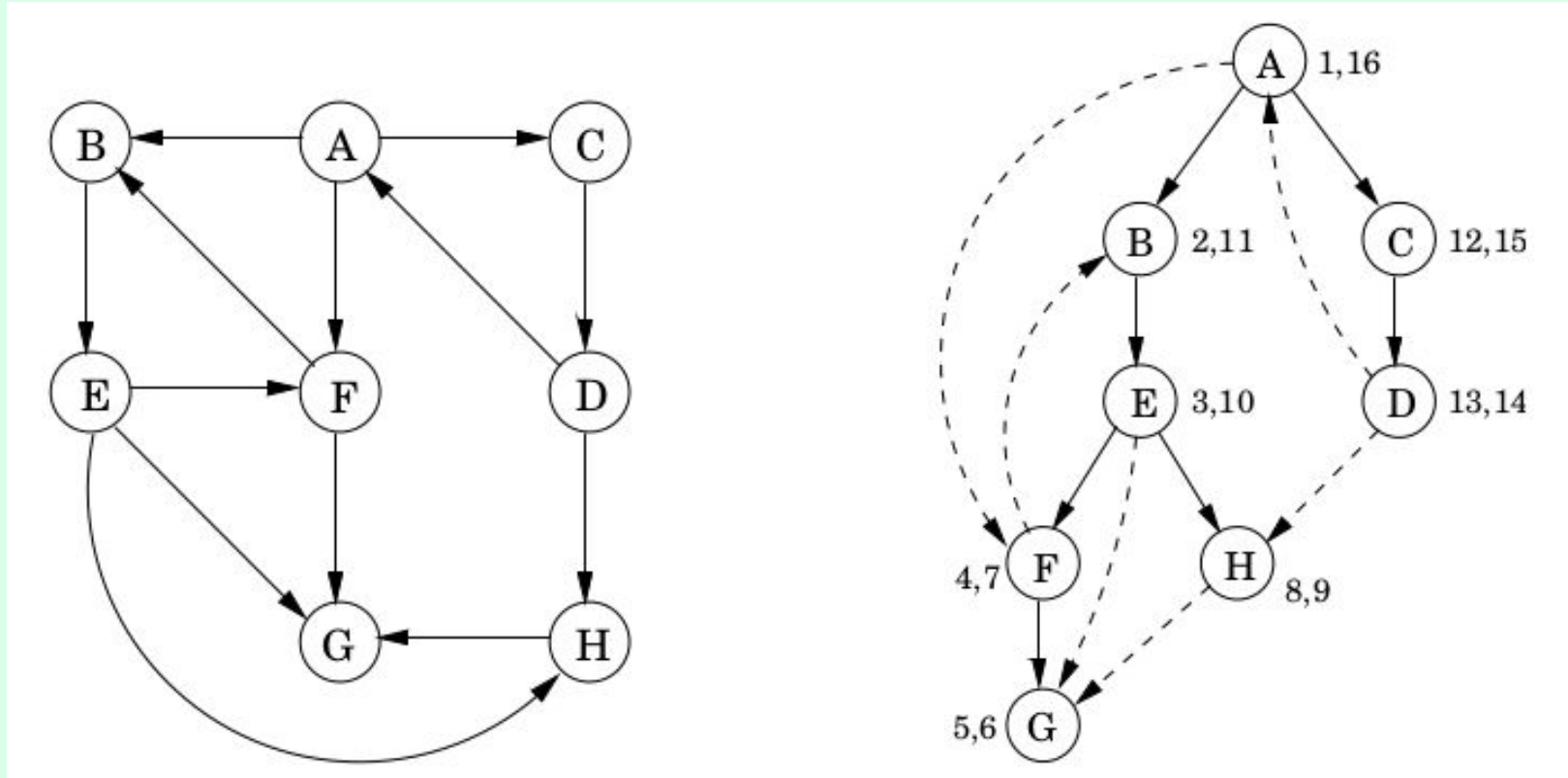
Email virus infection problem!

DFS in a directed graph

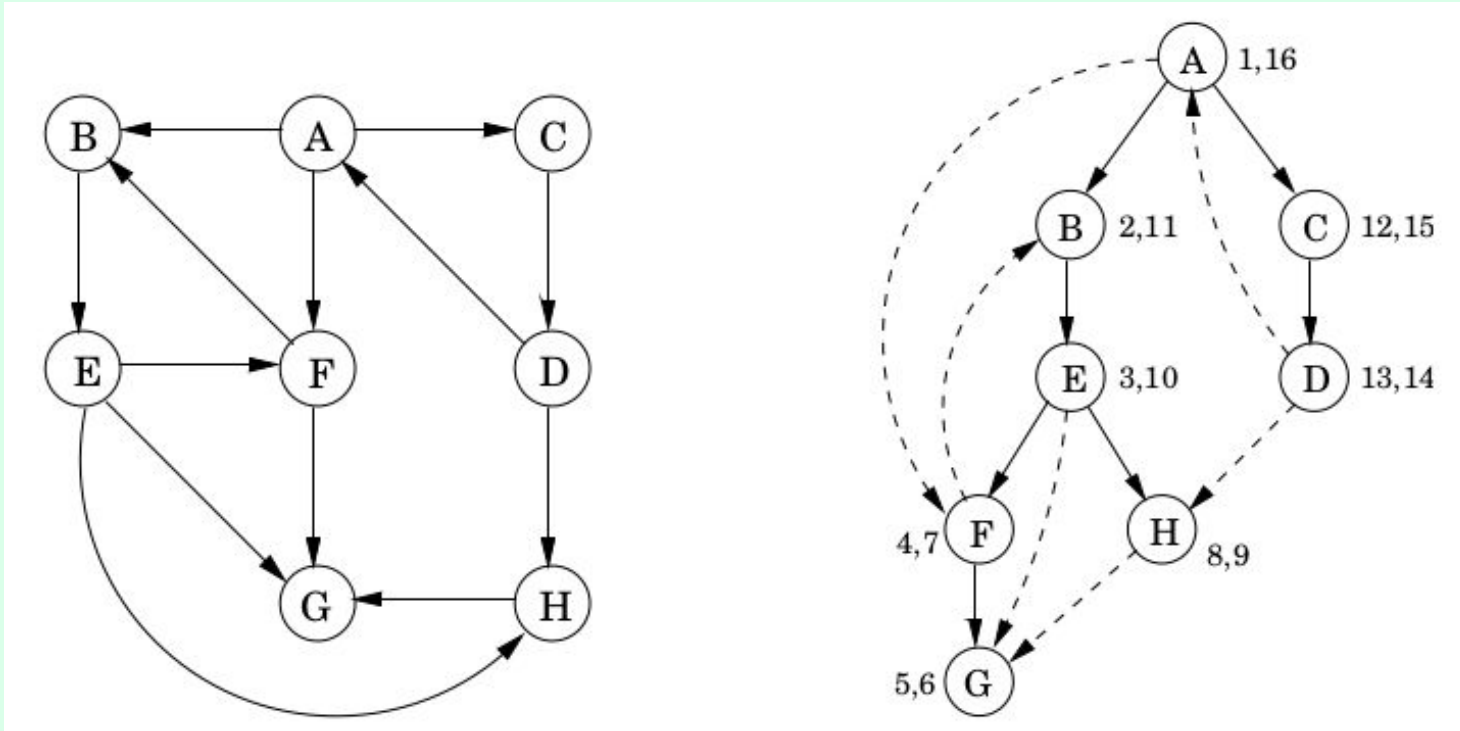


Start from A

DFS in a directed graph

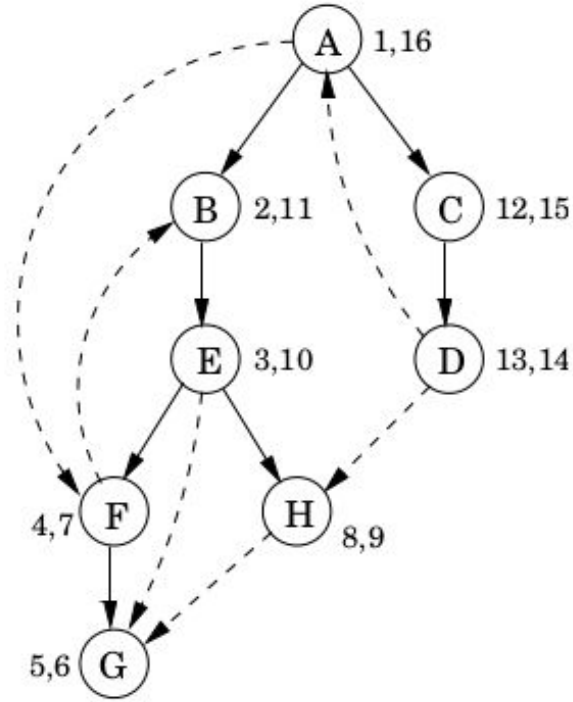
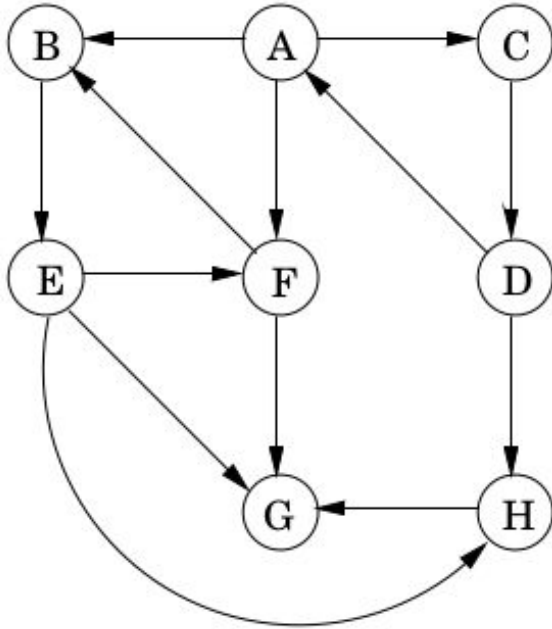


DFS in a directed graph



Pre and post numbers indicate ancestor and descendant relationships

DFS in a directed graph



$\text{pre}(u) < \text{pre}(v) < \text{post}(v) < \text{post}(u)$

[[]]
u	v	v	u

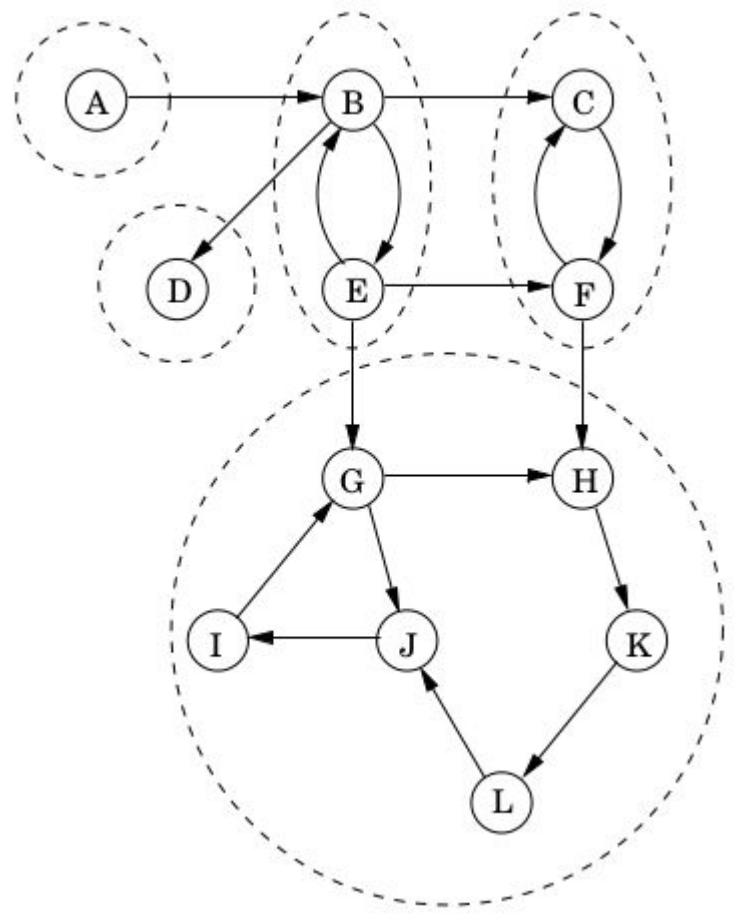
Email virus infection problem!

DFS in a directed graph

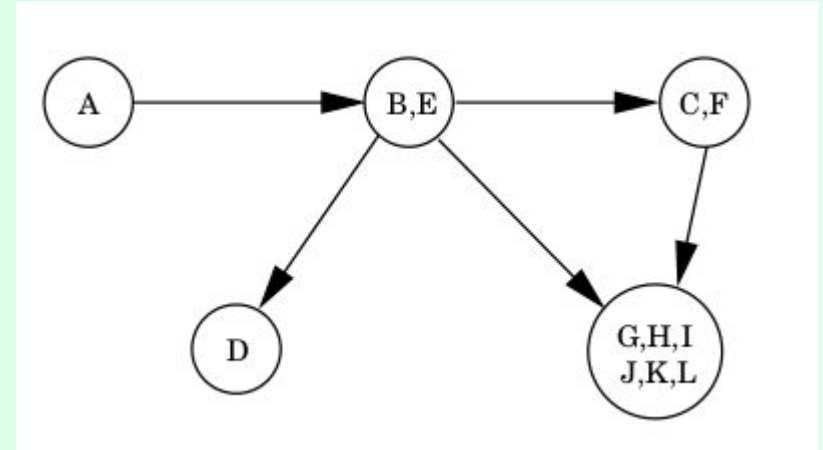
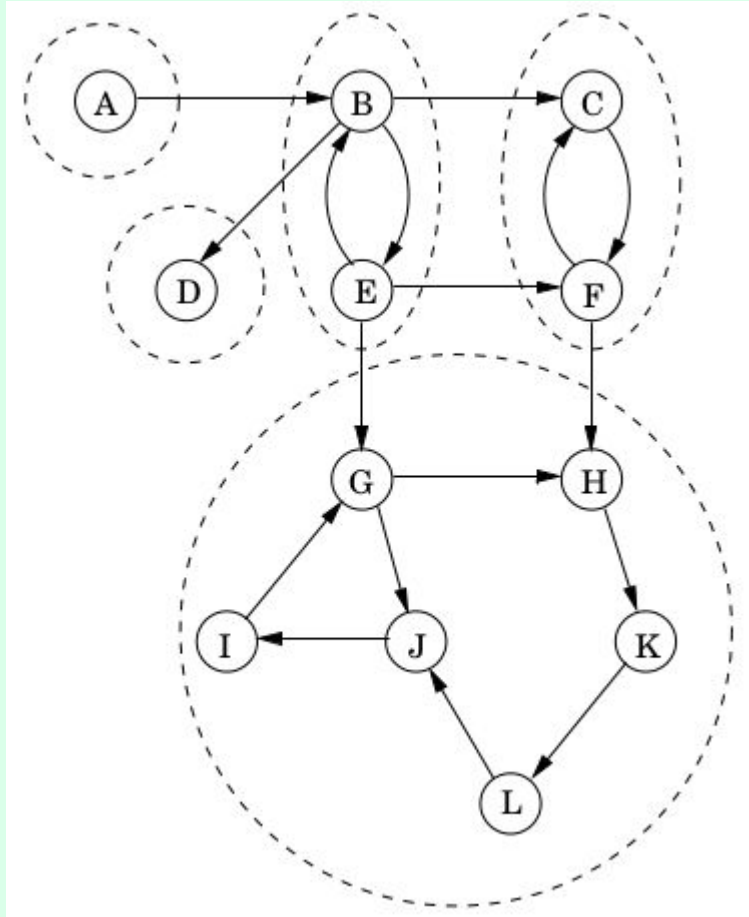
In a DAG, every edge leads to a vertex with a lower post number.

How will you find topological sort using DFS?

Is this directed graph strongly connected?



Directed graph with five strongly connected components



The resulting meta-graph **must be a DAG**.

Strongly Connected Components

The node that receives the **highest post number** in a depth-first search **must lie in a source strongly connected component**.

If C and C_0 are strongly connected components, and there is an edge from a node in C to a node in C_0 , then the highest post number in C is bigger than the highest post number in C_0 .

Decomposition of a directed graph into its strongly connected components

1. Run depth-first search on G^R .
2. Run the undirected connected components algorithm on G , and during the depth-first search, **process the vertices in decreasing order of their post numbers** from step 1.

Reference reading:

Algorithms by Dasgupta, Papadimitriou, and Vazirani

Depth-first search in directed graphs §3.3

Strongly Connected Components §3.4