Homework 5

1. DFS and strongly connected components
Show how the 2-phase DFS-based algorithm for computing strongly connected components works on the directed graph $G$ shown below; assume vertices are listed in alphabetical order in each adjacency list, and ties are broken alphabetically. Specifically, show the finishing times after phase 1, and the order in which components are output in phase 2. Draw the component graph of $G$, and clearly list the components.

![Diagram of a directed graph](image)

2. Topological sort
Consider the following algorithm for computing a topological sort of a DAG $G$: add the vertices to an initially empty list in non-decreasing order of their indegrees. Either argue that the algorithm correctly computes a topological sort of $G$, or provide an example on which the algorithm fails.

3. Miscellaneous
   (a) Can the number of strongly connected components of a graph decrease if a new edge is added? Why or why not? Can it increase? Why or why not?
   (b) What is the minimum number of strongly connected components that a directed acyclic graph (DAG) on $n$ nodes can have? What is the maximum number? Justify your answers.
   (c) Suppose you are given a directed graph $G$ on 6 vertices that has 3 strongly connected components (SCCs). What is the minimum number of SCCs $G$ can have after a new edge is added? Is it always possible to achieve this minimum for every such graph? Provide suitable examples to justify your answers.
   (d) Draw a directed acyclic graph on 5 vertices $\{a, b, c, d, e\}$ with exactly 2 topological sorts or argue that none exists. In the first case, list the 2 valid orders.
   (e) Draw a directed acyclic graph on 5 vertices $\{a, b, c, d, e\}$ with exactly 3 topological sorts or argue that none exists. In the first case, list the 3 valid orders.
Other suggested exercises (for practice; you do not need to turn in)

4. DFS

Give a counterexample to the conjecture that if there is a path from \( u \) to \( v \) in a directed graph \( G \), and if \( d(u) < d(v) \) in a DFS of \( G \), then \( v \) is a descendant of \( u \) in the DFS forest produced.

5. Topological sort

Prove or disprove: If a directed graph \( G \) contains cycles, then \( \text{TOPOLOGICAL-SORT}(G) \) produces a vertex ordering that minimizes the number of “bad” edges that are inconsistent with the ordering produced.