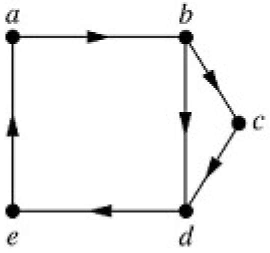
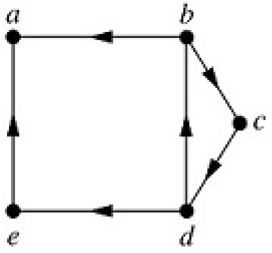
CSCI 270 Lecture 4: Graph Algorithms

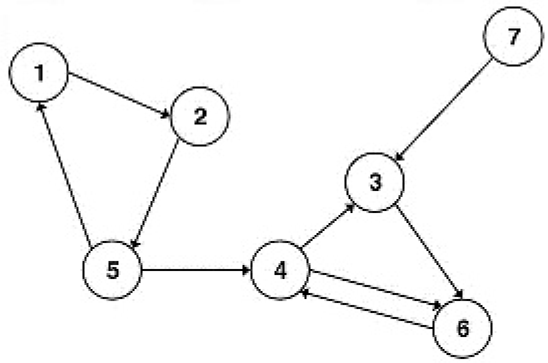
A directed graph is **strongly connected** if every pair of nodes can reach each other.

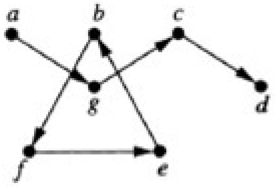
A directed graph is **connected** if for every pair of nodes, one can reach the other.

A directed graph is **weakly connected** if every pair of nodes can reach each other if you ignore edge directions.







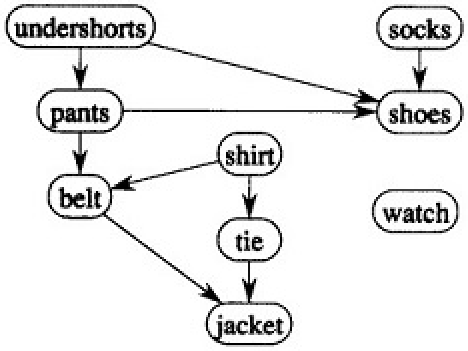


Connectivity Questions:

- Is the following statement true or false? Let s be an arbitrary node. G is strongly connected iff every node is reachable from s and s is reachable from all nodes.
- How could one write an algorithm to test whether a graph is strongly connected?
- Can we improve the efficiency of our algorithm?

A **directed acyclic graph** (or DAG) is a directed graph with no directed cycles.

A **topological order** of a directed graph is an ordering of its nodes $v_1, ..., v_n$ such that for every edge (v_i, v_j), $i < j$.



Questions:

- How are these two concepts related?
- Does every DAG have a topological ordering?
- True or False: If G is a DAG, then it has a node with no incoming edges.
- Design an algorithm to find a topological ordering, given the above idea.
- What is the runtime of the algorithm?
- What would this algorithm look like if we calculated the topological order in reverse?
- What basic algorithm would aid in finding the topological order in reverse?

Extra Problems:

- Chapter 3, exercises 2, 4, 5, 7, 9, 10
- Challenge problems: Chapter 3, exercises 6, 12