

Name: _____

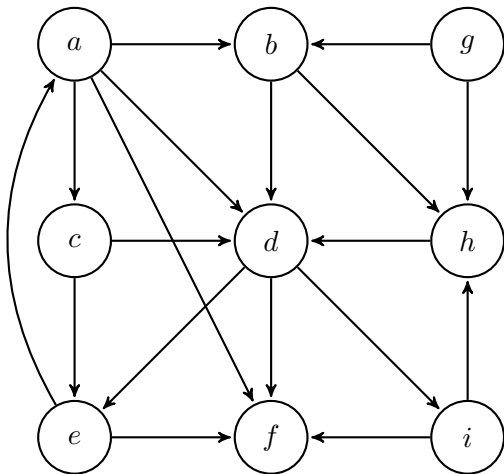
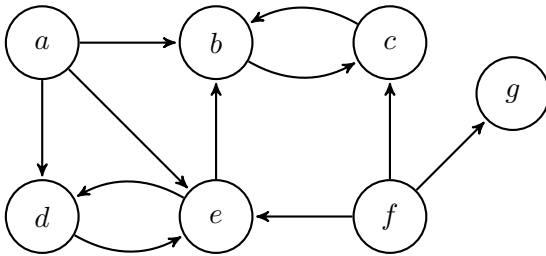
CSSE/MA 473 Worksheet for Class #7
December 8, 2022

Depth-First Search (DFS) Graph Traversal

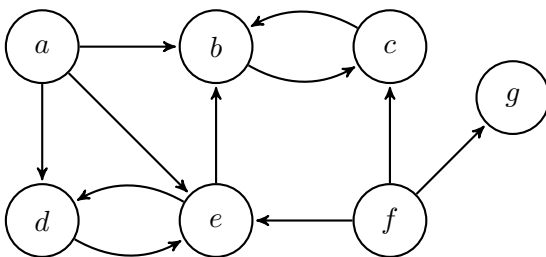
Conventions for this worksheet. To get consistent results, choose starting vertex and neighbor order alphabetically. For DFS traversals, write two step numbers in each node: “step when pushed onto call stack” and “step when popped from call stack.” Denote edges traversed by the graph search (or tree edges) with solid lines, and all other edges with dotted lines. Un-traversed edges (non-tree edges) come in three varieties:

- Back (B) Edge: to direct ancestor
- Forward (F) edge: to direct descendant
- Cross (C) edge: no direct ancestry

DFS Exercises. Traverse each graph with DFS, marking vertices and edges appropriately.



BFS Exercises. Traverse the graph with BFS, marking vertices with the “step when added to queue” and labelling edges appropriately.



Observations.

- If we think of the push/pop times as open/close parentheses, then DFS forms properly nested parentheses!
- A graph has no directed cycles (it is a dag) iff a DFS yields no _____ edges.
- In each, think: why are there none of the other two types of edges?
 - A DFS of an *undirected* graph will only yield _____ edges and _____ edges.
 - A BFS of an *undirected* graph will only yield _____ edges and _____ edges.

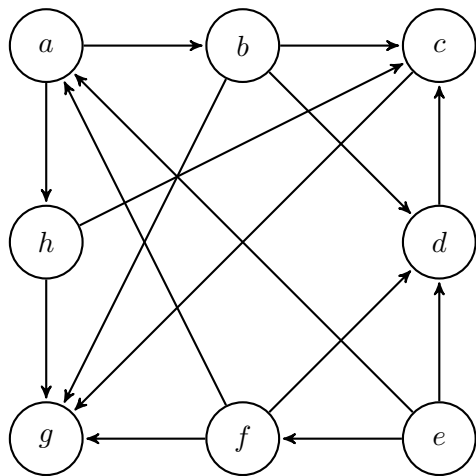
Topological Sort

Problem: Given a directed acyclic graph (dag), order the vertices so that for all edges (i, j) , i is before j .

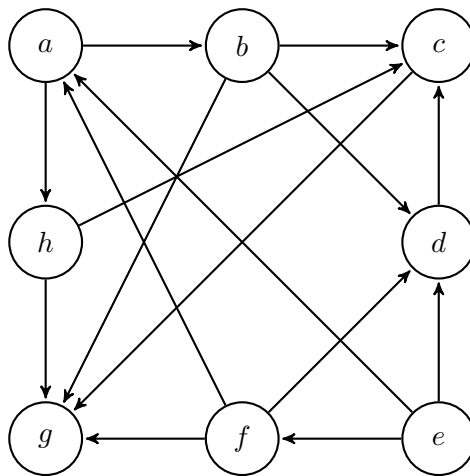
Algorithm 1. DFS-based. Run the DFS on the graph and output the vertices in reverse order of finishing time.

Algorithm 2. Source-removal. Iteratively remove a “source” (a vertex with in-degree 0) from the graph. Removal order forms a topological sort.

Exercise. Run both algorithms on the following graph. Are the results the same, or different?



DFS-based



Source-removal