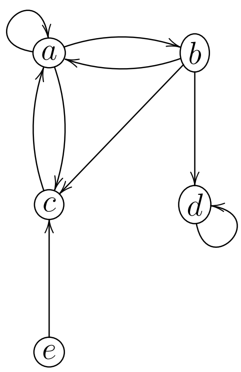
Exercise Sheet 14

Handout: December 19th — Deadline: December 26th, 4pm

Question 14.1 (0.25 marks)

Perform a depth-first search on the following graph visiting nodes in alphabetical order. Assume that all adjacency lists are sorted alphabetically. Write down the timestamps and the π-value of each node.

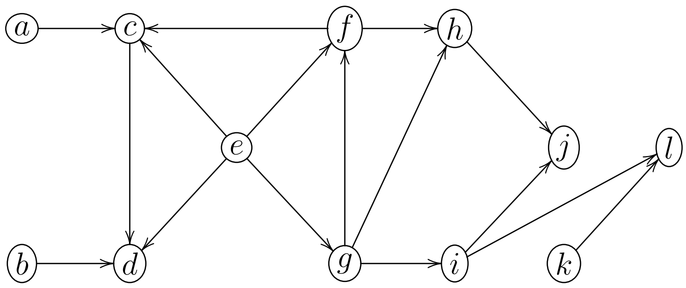


Question 14.2 (0.5 marks))

Prove or refute the following claim: if some depth-first search on a directed graph yields precisely one back edge, then all depth-first searches on this graph yield precisely one back edge.

Question 14.3 (0.25 marks)

Run Topological-Sort on the following directed acyclic graph. Assume that depth-first search visits nodes in alphabetical order and that adjacency lists are sorted alphabetically.



Question 14.4 (0.5 marks)

Recall from the lecture that DFS can be used to check whether a directed graph $G = (V, E)$ is acyclic or not, and that DFS runs in time $\Theta(|V| + |E|)$.

Give an algorithm that checks whether or not an *undirected* graph $G = (V, E)$ is acyclic and that *runs in time only* $O(|V|)$.

Question 14.5 (1 mark)

Implement TOPOLOGICAL-SORT(G) for a given directed graph $G(V, E)$. The algorithm should return a topological sort if the graph is acyclic or that no topological sort exists if the graph contains a cycle. The input will be:

- first line: N M (the number of vertices and edges).
- M lines each containing a pair $v_i v_j$ meaning there is an edge $v_i \rightarrow v_j$.

You have to first build the adjacency list representing the graph with the required attributes (colour, .d, .f .π).

The algorithm should run in time $O(V + E)$.