1. Graph traversal

(a) Consider the following graph. Suppose we want to traverse it, starting at node A.

If we traverse this using breadth-first search, what are two possible orderings of the nodes we visit? What if we use depth-first search?

(b) Same question, but on this graph:

2. Checking for Cycles

Given a graph as an adjacency list, design an algorithm that checks whether the graph has a cycle or not. The runtime of the algorithm must be $O(|V| + |E|)$, where $V$ is the vertex set and $E$ is the edge set. You may assume that the graph is unweighted, has no parallel edges and has no self loops.

3. Simulating Dijkstra’s

(a) Consider the following graph:
Suppose we run Dijkstra’s algorithm on this graph starting with vertex \( s \). What are the final costs of each vertex and the shortest paths from \( s \) to each vertex?

(b) Here is another graph. What are the final costs and shortest paths if we run Dijkstra’s starting on node \( A \)?

4. **Topological Sort**

Find a topological sort of the following graph: