1. What is more general, an algorithm for directed or for undirected graphs?

2. Apply Dijkstra’s shortest path algorithm to the following undirected graph, with start node A.

![Graph Diagram]

(You don’t need to make the underlying heap explicit.)

3. Suppose Dijkstra’s shortest path algorithm would terminate when the heap $H$ contains one node (instead of when it has become empty).

Would the algorithm then still be correct?

4. Given a directed weighted graph that consists of a cycle of three nodes and three edges with weight $-1$. Show how the Bellman-Ford algorithm detects this cycle.
5. Compute a minimum spanning tree in the following undirected weighted graph:

(a) Apply Prim’s algorithm, starting in node \( s \).
(b) Apply Kruskal’s algorithm.

In both cases, give one possible scenario. (You don’t need to make the underlying heap or disjoint-set data structure explicit.)

6. Argue that the worst-case time complexity of Prim’s algorithm, implemented using a heap, on a weighted graph with \( n \) nodes and \( m \) edges, is \( O(m \cdot \log n) \).

7. Explain why it is crucial for the worst-case time complexity of Kruskal’s algorithm that in the disjoint-set data structure, when two disjoint sets are joined by the union-find algorithm, the larger set subsumes the smaller set.
What would be the worst-case time complexity of Kruskal’s algorithm if the smaller set could subsume the larger set?

8. Argue that if all edges in an undirected weighted graph carry a different weight, then the graph has a unique minimum spanning tree.