Advanced Data Structures and Algorithms

Comprehensive Assignment Solutions

Arya Raag
Roll No: A125002

M.Tech (Computer Science and Engineering)

January 5, 2026

1 Graph Algorithms

Question 2. Explain why Dijkstra's algorithm cannot be applied to graphs with negative edge weights.

Detailed Solution:

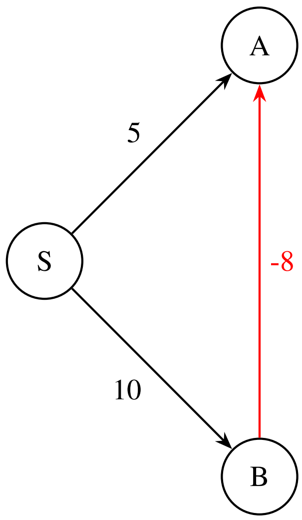
The Greedy Choice Property

Dijkstra's algorithm relies fundamentally on a greedy strategy. It maintains a set of "visited" or "finalized" nodes for which the shortest path from the source is known. The core assumption is: *"Once a node u is added to the finalized set, its shortest path distance $d[u]$ is optimal and will never change."*

This assumption holds for non-negative weights because extending a path can only increase (or keep constant) the total distance. You can never find a "shortcut" to a finalized node by going through a new, longer path.

Failure Mode with Negative Edges

When negative edges exist, this assumption collapses. Consider the following graph: $S \to A$ (cost 5), $S \to B$ (cost 10), $B \to A$ (cost -8).



Dijkstra's Execution Trace:

1. Start at S. Neighbors are $A(5)$ and $B(10)$.
2. Dijkstra picks the smallest distance: A with distance 5. It **finalizes** A.

3. It then processes B (distance 10). From B, it sees an edge to A with weight -8.

4. New path to A via B: $10 + (-8) = 2$.

5. This is smaller than the "finalized" distance of 5.

Conclusion

Dijkstra's algorithm finalized A prematurely. It does not go back to re-process finalized nodes. Consequently, it computes incorrect shortest path distances. For graphs with negative weights, the **Bellman-Ford algorithm** must be used, which can handle negative edges and detect negative cycles.
