### **Multiple Choice Questions (MCQs)**

1. **Which of the following graph algorithms is typically used to find the shortest path in a weighted graph?**
   * A) Depth-First Search (DFS)
   * B) Breadth-First Search (BFS)
   * C) Dijkstra’s Algorithm
   * D) Topological Sort
2. **In an interactive graph visualizer, which of the following features would be essential to visualize Dijkstra’s algorithm effectively?**
   * A) Displaying nodes with their traversal order only
   * B) Assigning weights to edges and updating shortest paths in real-time
   * C) Displaying only BFS and DFS traversals
   * D) Ensuring all nodes are connected directly to each other
3. **If a node has no incoming or outgoing edges, how will it typically appear in an interactive graph visualizer?**
   * A) As an isolated node with no connections
   * B) As a node with connections to all other nodes
   * C) As a hidden node
   * D) It will be removed automatically
4. **When using a visualizer for Breadth-First Search (BFS) traversal, what is the primary characteristic of the traversal order?**
   * A) Nodes are visited in a depth-first manner
   * B) Nodes are visited level-by-level, starting from the source
   * C) Nodes are visited based on edge weights
   * D) Nodes are visited randomly
5. **Which of these is a benefit of visualizing graph algorithms through an interactive tool?**
   * A) It eliminates the need to learn algorithms manually
   * B) It simplifies understanding the sequential execution of complex algorithms
   * C) It guarantees finding the optimal solution for every graph
   * D) It replaces the need for theoretical understanding of algorithms

**Answer:**

1. **Answer:** C) Dijkstra’s Algorithm
2. **Answer:** B) Assigning weights to edges and updating shortest paths in real-time
3. **Answer:** A) As an isolated node with no connections
4. **Answer:** B) Nodes are visited level-by-level, starting from the source
5. **Answer:** B) It simplifies understanding the sequential execution of complex algorithms

### **Logical Questions**

1. **If a graph visualizer allows a user to add edges with negative weights, what potential issues could arise when using Dijkstra’s algorithm?**
   * **Answer:** Dijkstra’s algorithm assumes all weights are non-negative. If negative weights are allowed, it could lead to incorrect shortest path calculations, as Dijkstra’s algorithm does not account for paths that might reduce distance by traversing a negative edge. This would make the algorithm unreliable in finding the true shortest path.
2. **Imagine you have implemented both BFS and DFS in your graph visualizer. Explain how you would visually demonstrate the difference between these algorithms.**
   * **Answer:** In the visualizer, BFS would show nodes being visited level by level, expanding outward from the source node in layers. DFS, on the other hand, would show a single path being explored deeply until it reaches the end before backtracking. To demonstrate, one could highlight the nodes in the order they are visited and use different colors or markers to emphasize the difference in traversal patterns.
3. **How could you use colors to enhance the visualization of Dijkstra’s algorithm in an interactive graph visualizer?**
   * **Answer:** Different colors could be used to indicate various states of each node. For example, unvisited nodes could be colored grey, nodes being evaluated (current shortest path candidates) could be colored yellow, and nodes with confirmed shortest paths could be colored green. Edge colors could also change as the shortest path tree develops, helping the user easily track the path-building process.
4. **Suppose a user wants to visualize a graph with cycles using DFS. What considerations would be necessary to handle cycles in the visualization, and how might this differ from a visualization of a tree-like graph?**
   * **Answer:** In a graph with cycles, DFS could potentially revisit nodes, leading to infinite loops if not handled correctly. To address this, the visualizer would need to mark visited nodes and avoid re-processing them, distinguishing them clearly from unvisited nodes. In a tree, this marking would not be necessary since trees have no cycles, allowing a straightforward traversal.
5. **If you are designing a feature that highlights the shortest path from a source node to a target node after running Dijkstra’s algorithm, how would you represent the distance information in the visualizer for user clarity?**
   * **Answer:** The shortest path could be highlighted in a unique color, such as blue, to make it distinct from other paths. Additionally, each node along this path could display a label or tooltip showing the cumulative distance from the source node. Edge weights along the path could also be bolded or displayed in a larger font to emphasize the calculations involved in determining the shortest path.