1. True or \(\boxed{False}\) (Circle one): If all edge weights are one, and the graph is undirected, then the single-source shortest path problem can be solved by performing a depth-first-search starting from the source vertex \(s\).

2. \(\boxed{True}\) or False (Circle one): If all edge weights are one, and the graph is undirected, then the single-source shortest path problem can be solved by performing a breadth-first-search starting from the source vertex \(s\).

3. The single-source shortest path problem is only defined (i.e., makes sense) for weighted graphs which do not have \(\boxed{\text{negative weight cycles}}\). Assuming the single-source shortest path problem is defined for a given graph, Dijkstra’s shortest path tree algorithm is only guaranteed to work correctly if the graph is further restricted to \(\boxed{\text{positive weighted graphs}}\).

4. Dijkstra’s algorithm is similar to \(\boxed{\text{Prim-Jarnik’s}}\) algorithm for computing minimum spanning trees and runs in \(O((n + m) \log n)\) time.

5. True or \(\boxed{False}\) (Circle one): Consider the graph shown below. Dijkstra’s algorithm can be used to solve the single-source shortest path problem on this graph, and the algorithm will run in \(O(n^2)\) time.

![Graph Diagram]