Prims Algorithm:

Prim’s algorithm is used to find the Minimum Spanning Tree for a given graph. It’s starts with a empty spanning tree and works its way through several adjacent nodes, exploring all of the connected edges along the way. Edges with the minimum weights that do not cause cycles in the graph get selected for the final MST and hence its called a greedy algorithm.

When considering a weighted graph that contains negative weights , prims does work fine because in Prim’s algorithm the tree evolves by adding the least weight edge that connects a tree vertex to a non-tree vertex. The edge selection is not effected by negative weights.

This Algorithm works by starting at a random node and traversing the graph selecting the edge with the lowest weight and hence every node should be accessible from every node present in the graph but when the graph is a directed graph , every node is not reachable from every other node but prims algorithm considers that every node is connected and hence there is a contradiction . so prims does not provide an optimal result for the given directed graph.

Kruskal’s Algorithm:

Kruskal's algorithm is a greedy algorithm in graph theory that is used to find the Minimum spanning tree. Kruskal’s algorithm sorts all the edges in increasing order of their edge weights and keeps adding nodes to the tree only if the chosen edge does not form any cycle. Also, it picks the edge with a minimum cost at first and the edge with a maximum cost at last. Every 2 nodes are taken and the path with the minimum cost is chosen for this algorithm.

When taking a negative weights included graph , Kruskal’s algorithm sorts the edges according to weights, which doesn’t change when negative weights are present and chooses the best edge from it each time.

In kruskal’s , after each step the graph is checked for cycles But Kruskal’s algorithm fails to detect the cycles in a directed graph as there are cases when there is no cycle between the vertices but Kruskal’s Algorithm assumes it to cycle and don’t take consider some edges due to which Kruskal’s Algorithm fails for directed graph.

Dijkstra’s Algorithm:

Dijkstra’s Algorithm works on the basis that any subpath of the shortest

path between vertices A and D is also the shortest path between vertices B and D.

Dijkstra used this property in the opposite direction i.e., we overestimate the

distance of each vertex from the starting vertex. Then we visit each node and its

neighbours to find the shortest subpath to those neighbours.

The algorithm uses a greedy approach in the sense that we find the next best

solution hoping that the end result is the best solution for the whole problem.

The problem with Dijkstra’s algorithm is that it is believed that all costs in the given

graph are non-negative, so adding any positive number on a vertex that has already

been visited will never change its minimality.

Since Dijkstra follows a Greedy Approach, once a node is marked as visited it cannot be reconsidered even if there is another path with less cost or distance. This issue arises only if there exists a negative weight or edge in the graph. So, this algorithm fails to find the minimum distance in case of negative weights.

Conclusion:

All the above three algorithms don’t provide optimal solution for the given graph and hence to provide a possible and optimal result Bellman ford algorithm can be used . The Bellman–Ford algorithm is an algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph. It is slower than Dijkstra&#39;s algorithm for the same problem, but more versatile, as it is capable of handling graphs in which some of the edge weights are negative numbers. Assuming there are negative edges in the graph, but there are no negative cycles, the Bellman-Ford algorithm iterates through all the edges multiple times (V-1 times), irrespective of the fact whether the vertices are visited or not, thus resulting in successfully finding the optimal low-cost path.

I have used C programming language to code all the algorithms as it is the language I used previously to code these algorithms and I find it easier to understand.