Dijkstra’s algorithm does not work if edges have negative weights. Consider the following idea to rectify this:

1. Let $W$ be the smallest negative weight of any edge ($W=0$ if no edge has negative weight).
2. Add $|W|$ to the weight of every edge, thereby making all edge weights non-negative.
3. Run Dijkstra’s algorithm to find shortest paths based on the new non-negative weights.
4. The resulting shortest paths are also shortest paths in the original graph before the adjustment of the weights in Step 2.

Does this work? Prove or disprove.
Here is Dijkstra’s algorithm:

\[ R := \text{empty set} \]
\[ d(s) := 0; \ p(s) = \text{nil} \]
\[ \text{for each node } v \neq s \ \text{do} \ d(v) := \infty \]
\[ \text{while } R \neq V \ \text{do} \]
\[ \quad u := \text{node not in } R \text{ with minimum } d\text{-value} \]
\[ \quad R := R \cup \{u\} \]
\[ \quad \text{for each node } v \ \text{s.t. } (u,v) \text{ is an edge do} \]
\[ \quad \quad \text{if } d(v) > d(u) + \text{wt}(u,v) \ \text{then} \]
\[ \quad \quad \quad d(v) := d(u) + \text{wt}(u,v); \ p(v) := u \]

Does the algorithm work if we skip the last iteration of the loop (i.e., \( R \neq V \) is replaced by \(|R| < |V| - 1\))?
Question 3

Suppose that nodes, in addition to edges, have weights. The weight of a path is now the sum of the weights of the edges AND nodes on the path, including both endpoints.

Assuming all node and edge weights are non-negative, give an efficient algorithm that finds minimum weight paths from a given node to all nodes in a directed graph.
Question 4

Suppose we are given a directed graph with non-negative edge weights, and we want to find the maximum weight path from a given node to all nodes.

If the graph has cycles, the problem is not well defined because the more times we go around a cycle, the longer the path.

It makes sense, however, to look for a maximum weight simple path, i.e., a path where no node is ever repeated.
Does the following modification of Dijkstra’s algorithm find max weight simple paths from a given node s to all nodes?

(Changes shown in red.)

R := empty set
d(s) := 0; p(s) = nil
for each node v ≠ s do d(v) := −∞
while R ≠ V do
  u := node not in R with maximum d-value
  R := R U {u}
  for each node v s.t (u,v) is an edge do
    if d(v) < d(u)+wt(u,v) then
      d(v) := d(u)+wt(u,v); p(v) := u
Question 5

Dijkstra’s algorithm applied to a connected undirected graph G, starting at some node s, constructs a shortest path tree.

1. Why is this a spanning tree of G?

2. Is it a minimum spanning tree of G?