Dijkstra’s algorithm is given below.

(1) The algorithm is applied to a weighted directed acyclic graph, where the weights of the edges are positive. Will the algorithm find the weights of the shortest paths correctly? What is the complexity of the algorithm when it is applied to such a weighted directed acyclic graph? Explain your answers briefly.

(2) Repeat part 1 assuming that the graph may have edges with negative weights.

Dijkstra(G,w,s)

1 Initialize(G,s)
2 S ← φ
3 Q ← V[G]
4 while Q ≠ φ do
5 u ← Extract−Min(Q)
6 S = S ∪ {u}
7 for each vertex v ∈ Adj[u] do
8 Relax(u,v,w)

Solution:

(1) Yes: The algorithm is correct for all directed graphs with non-negative weights.
   Complexity: $O(|V|^2)$. The while loop performs $O(|V|)$ iterations. Extract-Min is $O(|V|)$.

(2) No: In the following graph, after $s$ is selected and relaxation is applied, $d[A] = 4$ and $d[B] = 3$. Dijkstra’s algorithm will select $B$, and assign $d[C] = 4$. This label will not change since $B$ will not be considered again. However, the weight of the shortest path to $C$ is 3, through $A$.

Complexity: $O(|V|^2)$ for the same reason as in part 1.