1 Consider the following input to the single-source shortest path problem, with starting vertex s and with each edge labeled with its length. $V = \{s, v, w, t\}$ and edge set $E = \{(s, v, 1), (s, w, 4), (v, w, 2), (v, t, 6), (w, t, 3)\}$.

1. $0, 1, 2, 3$
2. $0, 1, 3, 6$
3. $0, 1, 4, 6$
4. $0, 1, 4, 7$

(b). Let $\delta(v)$ be the weight of the shortest path starting from the source vertex s and going to the vertex v for all $v \in V$. $\delta(s) = 0$, $\delta(v) = 1$,$\delta(w) = 3$ and $\delta(t) = 6$.

2 Which of the following running times best describes a straightforward implementation of Dijkstra's algorithm for graphs in adjaceny-list representations? As usual, n and m denote the number of vertices and edges respectively, of the input graph.

1. $O(m + n)$
2. $O(m \cdot \log n)$
3. $O(n^2)$
4. $O(m \cdot n)$

(4) Each iteration takes $O(m)$ time and we will have at most $O(n)$ iterations, hence the overall running time is $O(m \cdot n)$.

3 Consider a directed Graph G with distinct and non-negative edge lengths. Let s be a starting vertex and t a destination vertex and assume G has at least one $s - t$ path. Which of the following statements are true?

1. The shortest $s - t$ path might have as many as $n - 1$ edges, where n is the number of vertices.
2. There is a shortest $s - t$ path with no repeated vertices
3. The shortest $s - t$ path must include the minimum-length edge of G.
4. The shortest $s - t$ path must exclude the maximum-length edge.

(1), (2) are correct. (1) is correct because if for example the graph was a tree and tree be a "line" graph with the starting vertex one leaf and the target vertex another leaf then we would need $n - 1$ edge to go from the source to the target vertex. The above reasoning also shows that (2) is correct. (3) and (4) can be shown to be incorrect with counterexamples.

4 Consider a directed graph G with a starting vertex s, a destination t, and nonnegative edge lenghts. Under what conditions is the shortest $s - t$ path guaranteed to be unique?

1. When all edge lengths are distinct positive integers.
2. When all edge lengths are distinct powers of 2.
3. When all edge lengths are distinct positive integers and the graph G contains no directed cycles.
4. None of the other options are correct

(2) is correct because we can use a binary representation to encodes the path and since every edge length is an distinct power of 2 then no two numbers will have the same binary representation. $(1), (2)$ can be shown to be false by counter-examples.

5 Consider a directed graph G with nonnegative edge lengths and two distinct vertices, s and t.Let P denote a shortest path from s to t. If we add 10 to the length of every edge in the graph, then

1. P definitely remains a shortest $s - t$ path.
2. P definitely does not remain a shortest $s - t$ path.
3. P might or might not remain a shortest $s - t$ path.
4. If P has only one edge, then P definitely remains a shortest $s - t$ path.

(3) and (4) are correct. (1) and (2) can be shown to be incorrect by counterexamples validating the claim of (3). (4) is correct because any other canditate path P' must have length ≥ 1. If the length of P' is exactly 1 then it must be larger than P since if not then it would have been the $s-t$ path for the original graph. If the length of $P's$ is greater than 1 than it must larger than P since every extra edge penalizes the path by 10 and P is only penalized by 10 once.

6 Consider a directed graph G and a starting vertex s with the following properties: no edges enter the starting vertex s; edges that leave s have arbitrary lengths; and all other edge lengths are nonnegative. Does Dijkstra's algorithm correctly solve the single-source shortest path problem in this case?

1. Yes, for all such inputs.
2. Never, for no such inputs.
3. Maybe, maybe not

4. Only if we add that the assumption that G that contains no directed cycles with negative total length.

(1) is correct. We can show this by taking the largest negative edge weight incident to the source vertex s and adding the negation to all edge weights incident to the s. This will add the negation of the length of the most negative weight to all the shortest path to vertices hence preserving the original shortest path solution.

7 Consider a directed graph G and a starting vertex s. Suppose G has some negative edge lengths but no negative cycles, meaning G does not have a directed cycle in which the sum of its edge lengths is negative. Suppose you run Dijkstra's algorithm on this input. Which of the following statements are true?

1. Dijkstra's algorithm might loop forever.
2. It's impossible to run Dijkstra's algorithm on a graph with negative edge lengths.
3. Dijkstra's algorithm always halts, but in some cases the shortest-path distances it computes will not all be correct.
4. Dijkstra's algorithm always halts, and in some cases the shortest-path distances it computes will all be correct.

(3) and (4) are correct. (1) is incorrect because Dijkstra eliminates vertices that have been visited thus Dijkstra will enventually terminate and infact terminates in $n-1$ iterations where n is the number of vertices. (2) is incorrect because it is possible to run Dijkstra with negative edge lengths but correctness is not guaranteed.

8 Consider the previous problem, suppose now that the input graph G does not contains a negative cycle, and also a path from the starting vertex s to this cycle. Suppose you run $Dijkstra's$ algorithm on this input. Which of the following statements are true?

1. Dijkstra's algorithm might loop forever.
2. It's impossible to run Dijkstra's algorithm on a graph with a negative cycle.
3. Dijkstra's algorithm always halts, but in some cases the shortest-path distances it computes will not all be correct.
4. Dijkstra's algorithm always halts, and in some cases the shortest-path distances it computes will all be correct.

(2) is correct. If we encounter a negative cycle we can expect to find smaller and smaller paths thus resulting in an infinite loop. (3) and (4) are incorrect because dijkstra might not halt. (1) is incorrect because no description of the algorithm itself that relies on there being no negative cycle.

9 Consider a directed graph $G = (V, E)$ with non-negative edge lengths and a starting vertex s. Define the bottleneck of a path to be the maximum lengths of one of its edges. Show how to modify Dijkstra's algorithm to compute, for each vertex $v \in V$, the smallest bottleneck of any $s - v$ path. Your algorithm should run in $O(m \cdot n)$ time, where m and n denotes the number of edges and vertices, respectively.

Replace the lines of $len(v) + l_{vw}$ with $\max\{len(v), l_{vw}\}$ and $len(v^*) + l_{v^*w^*}$ with $\max\{len(v^*), l_{v^*w^*}\}$