COMP S265F Design and Analysis of Algorithms
Lab 9: Dijkstra's Algorithm – Suggested Solution

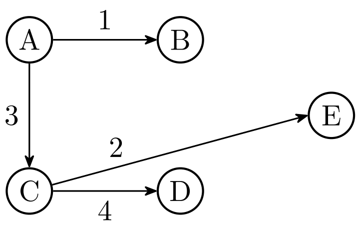
Question 1.

(a) In running the Dijkstra's algorithm starting at the source vertex A, for any vertex v, let $\pi[v]$ be the vertex such that $d[v]$ is updated by relaxing the edge $(\pi[v], v)$.

The following table shows $d[v]$ and $\pi[v]$ of all vertices v, and the set of vertices in the shortest path subtree P after each iteration of the while loop.

Iteration	Vertex v	A	B	C	D	E	P
1st	$d[v]$	0	1	3	∞	∞	$\{A\}$
	$\pi[v]$	–	A	A	–	–	
2nd	$d[v]$	0	1	3	∞	7	$\{A, B\}$
	$\pi[v]$	–	A	A	–	B	
3rd	$d[v]$	0	1	3	7	5	$\{A, B, C\}$
	$\pi[v]$	–	A	A	C	C	
4th	$d[v]$	0	1	3	7	5	$\{A, B, C, E\}$
	$\pi[v]$	–	A	A	C	C	
5th	$d[v]$	0	1	3	7	5	$\{A, B, C, D, E\}$
	$\pi[v]$	–	A	A	C	C	

(b) The last row of the table in (a) shows all edges $(\pi[v], v)$ in the shortest path tree obtained by the Dijkstra's algorithm:



Question 2. In the proof of correctness of the Dijkstra's algorithm in Unit 5 Slides 20-29, the restriction that edge weights are positive is used in the argument in Slide 24, which shows that the tree path from s to u and then to v is the shortest among all paths σ from s to v in the graph G.

In fact, this restriction is stronger than what is needed. Suppose $\sigma = (s = p_0, p_1, \ldots, p_m, q_1, q_2, \ldots, q_k = v)$ is a *shortest* path from s to v, where $p_0, p_1, \ldots, p_m$ are all in the shortest path subtree P of the Dijkstra's algorithm, and q_1 is the first vertex that is not in P.

We first show that we can assume that $q_i \neq s$ for any i. If there is a vertex $q_i = s$, then the path $(s, p_1, p_2, \ldots, p_m, q_1, q_2, \ldots, q_i = s)$ is a cycle. Since σ is a shortest path from s to v, the total weight of all the edges in this cycle must be negative or 0. As stated in the question, the graph does not contain any negative-weight cycle, so the total weight of all the edges in this cycle must be 0 and we can remove this cycle from σ to get another shortest path from s to v. Repeating this process allows us to remove all $q_i = s$. Thus, we can assume that $q_i \neq s$ for any i.

With the assumption that $q_i \neq s$ for any i, the weight of any edge in the path $(q_1, q_2, \ldots, q_k = v)$ must be non-negative as there is no edge leaving the source vertex s. Therefore, we still have the inequality

$$w(\sigma) = w((s = p_0, p_1, \ldots, p_m, q_1, q_2, \ldots, q_k = v)) \geq w((s = p_0, p_1, \ldots, p_m, q_1)) \ ,$$

which is at least $\delta(s, p_m) + w(p_m, q_1) = d(p_m, q_1) \geq d(u, v)$, i.e., the distance of the tree path from s to u and then to v. Therefore, the Dijkstra's algorithm still correctly finds shortest paths from s in this graph.