Problem 1 (10 points). Prove or give a counter-example for each of the following statements.

1. Let p be a shortest path from vertex s to vertex t in a directed graph. If the length of each edge in the graph is increased by 1, p will still be a shortest path from s to t.
2. Let p be a shortest path from vertex s to vertex t in a directed graph. If the length of each edge in the graph is decreased by 1, p will still be a shortest path from s to t.

Problem 2 (10 points). Consider the following implementation of priority queue *PQ* with an array *S*. *insert (PQ, x)*: add *x* to the end of *S*; *decrease-key (PQ, x, key)*: set the key of element *x* as *key*; *empty (PQ)*: check whether the size of *S* is 0; *find-min (PQ)*: traverse *S* and return the element with smallest key; *delete-min (PQ)*: first traverse *S* to locate element with smallest key, then remove this element by shifting all elements on its rightside.

1. Analyze the running time of each above operation.
2. What is the running time of Dijkstra's algorithm if this implementation of priority queue is used?

Problem 3 (20 points). Let $G = (V,E)$ be a directed graph with positive edge length. Let $t \in V$. Give an algorithm runs in $O(|V|^2)$ time for finding shortest paths between all pairs of nodes, such that these paths pass through t. *(Hint: use the results in Problem 2.)*

Problem 4 (20 points). Let $G = (V,E)$ be a directed graph with positive edge length. Design an algorithm runs in $O(|V|^3)$ to find the length of the shortest cycle in G.

Problem 5 (20 points). Given directed graph $G = (V,E)$ with positive edge length, vertex $s \in V$, describe how to modify Dijkstra's algorithm so that the algorithm also sets a binary array $unique[1 \cdots |V|]$, where $unique[u] = 1$ if there exists a unique shortest path from s to u, and $unique[u] = 0$ if there are more than one shortest paths from s to u.

Problem 6 (20 points). Let $G = (V,E)$ be a directed graph with positive edge length $l(e) > 0$ for any $e \in E$. For vertex $v \in V$ there is also an associated positive *vertex weight* $w(e) > 0$. For any path we define its *length* as the sum of the length of its all edges plus the sum of the weights of its all vertices. Given $s \in V$, design an algorithm runs in $O((|V| + |E|) \cdot \log |V|)$ for computing the shortest paths (in terms of this new definition of length) from s to all vertices.