Note: In the problems below, when you are asked to write a pseudocode, you can use any algorithm / procedure we learned in class as subroutines. Also, to help the TA understand your code, it will help to provide a description of your algorithm.

1. [10 points] Give the EDGES of the minimum spanning tree of the weighted graph in the figure below in the order they would be output by Prim’s algorithm starting at vertex $v_1$.

![Graph](image)

2. [10 points] We have proved in class that given a weighted graph $G = (V, E)$ with distinct edge weights, the minimum-weight edge $e_{\text{min}}$ must exist in the minimum spanning tree $T$ of $G$. When the edge weights are not distinct, then there are more than one minimum spanning trees for the input graph. Give an example of a graph $G = (V, E)$ and a specific minimum spanning tree $T^*$ of $G$ that satisfy the following conditions: (1) edge weights in $G$ are not distinct and multiple edges in $G$ have the minimum edge weight; and (2) at least one of the minimum weight is not in this minimum spanning tree $T^*$ of $G$. (That is, suppose the minimum edge weight is $w$ among the weights of all edges in $G$. Multiple edges have edge weight $w$. At least one of such min-weight edges is not in the minimum spanning tree $T^*$ you gave. You should provide this graph $G$ and this specific minimum spanning tree $T^*$.)

3. [10 points] Given a weighted graph $G = (V, E)$, suppose that all edge weights in $G$ are integers in the range from 1 to $W$ for some constant $W$. How fast can you make Prim’s algorithm run? Justify your answer briefly.

4. [10 points] (4.a) Give an example of a weighted undirected graph $G = (V, E)$ where its shortest path tree rooted at the source node $v_1$ equals to the minimum spanning tree of $G$. (Do not use the graph where all edges have the same edge weight.)

(4.b) Give an example of a weighted undirected graph $G = (V, E)$ where its shortest path tree rooted at the source node $v_1$ does not equal to the minimum spanning tree of $G$. (Do not use the example I have in the slide, nor the example I gave in class.)

5. [10 points] Give the EDGES of the shortest path tree of the weighted graph in Figure 5 in the order they would be output by Dijkstra’s algorithm starting at vertex $v_1$.

![Graph](image)
6. [8 points] Give an example of a weighted undirected graph $G = (V, E)$ that satisfies the following condition: If we perform the Dijkstra’s algorithm (introduced in class) on $G$ starting at source $v_1$, at the end of the algorithm, the $u.distance$ value is not correct (i.e., $u.distance \neq \delta(v_1, u)$) for at least one vertex $u$. (Note, this means that the input graph $G$ has to have negative edge weights, as otherwise, we have shown in class that Dijkstra’s Algorithm computes the correct shortest path distance to the source node $v_1$ for all nodes. You cannot use the example that I gave in class.)

7. [10 points] Given a shortest path (SP) tree $T$ of a weighted graph $G = (V, E)$ (with positive edge weights) with source node $v_s$ (i.e., $v_s$ is also root of the SP tree $T$), write the pseudocode of an algorithm that print the shortest path distance from $v_s$ to all other reachable nodes in $G$. Give the asymptotic running time complexity of your algorithm in terms of $n = |V|$ and $m = |E|$.

8. [7 points] Specify whether the following statement is True or False. If it is true, justify it. If it is false, disprove it by giving a counter example.

[Given a weight undirected graph $G = (V, E)$ with positive weights $w : E \rightarrow R$, let $T$ be a shortest path tree of $G$ from a source node $v_s \in V$. Now multiply the weight of every edge by 3. $T$ remains a shortest path tree of the graph $G$ (from source $v_s$) with new weights.]

9. [20 points] Suppose we are given a directed weighted graph $G = (V, E)$ modeling the road network between $n = |V|$ cities. Specifically, each node $v_i \in V$ represents a city. Assume all edge weights are positive. If there is an edge $e = (v_i, v_j) \in E$, then there is a road between city $v_i$ and $v_j$, and the weight $weight(e)$ represents the time it takes to go from city $v_i$ to $v_j$ on this road. We also refer to each edge in $E$ as one road segment. The total number of road segments (i.e., edges) is $m = |E|$.

(a) [Given a fixed city $v_s$, suppose Alex wants to travel to only cities that are within $k$ number of road segments. Write the pseudocode of an algorithm that outputs these cities. Give the running time of your algorithm. (You can use any algorithm we discussed in class as subroutines.)]

(b) [Given a fixed city $v_s$ and a threshold $\tau$, write the pseudocode of an algorithm that output the set of nodes that are within at most $\tau$ shortest distance away from the source $v_s$. Give the running time of your algorithm.]

10. [10 points] Suppose we are given an undirected graph $G = (V, E)$ with positive edge weights $\omega : E \rightarrow R$. The weight of a cycle is defined as the sum of weights of all edges in this cycle. Given an edge $e = (u, v)$, describe an efficient algorithm to compute a minimum-weight cycle $C^*$ that containing $e$: that is, $C^*$ is a cycle containing $e$, and among all cycles containing $e$, $C^*$ has the smallest weight. Give the time complexity of your algorithm. (Slower algorithm receives fewer points.)