# Homework Assignment #8

## Due Time/Date

1:20PM Tuesday, November 26, 2024. Late submission will be penalized by 20% for each working day overdue.

## How to Submit

Please write or type your answers on A4 (or similar size) paper. Put your completed homework on the table in front of the instructor's desk before the class on the due date starts. For early or late submissions, please drop them in Yih-Kuen Tsay's mail box on the first floor of Management College Building 2. You may discuss the problems with others, but copying answers is strictly forbidden.

## Problems

There are five problems in this assignment, each accounting for 20 points. (Note: problems marked with "(X.XX)" are taken from [Manber 1989] with probable adaptation.)

1. Consider Dijkstra's algorithm for single-source shortest paths. The values of $SP$ for all vertices may be stored in either an ordinary array or a min heap. How do these two implementations compare in terms of time complexity? Please explain.

2. (7.9) Prove that if the costs of all edges in a given connected graph are distinct, then the graph has exactly one unique minimum-cost spanning tree.

3. The well-known Kruskal's algorithm computes the minimum-cost spanning tree of a given connected weighted undirected graph with $n$ vertices as follows:

   Initially, it treats the $n$ vertices as a forest of $n$ trees, each of a single node. It then examines the edges one by one in *increasing order* of their weights. If the edge under examination connects two different trees (i.e., the edge does not complete a cycle), it is included in the forest (causing the forest to evolve, eventually becoming a single tree).

   Please present the algorithm in adequate pseudocode utilizing the two operations of the Union-Find data structure. What is the time complexity of the algorithm? Please explain.

4. What is wrong with the following algorithm for computing the minimum-cost spanning tree of a given weighted undirected graph (assumed to be connected)?

   > If the input is just a single-node graph (base case), return the single node. Otherwise (inductive step), divide the graph into two disjoint subgraphs arbitrarily (by removing the edges between the two subgraphs),

recursively compute their minimum-cost spanning trees, and then connect the two spanning trees with an (earlier removed) edge between the two subgraphs that has the minimum weight.

5. (7.61) Let $G = (V, E)$ be a connected weighted undirected graph and $T$ be a minimum-cost spanning tree (MCST) of $G$. Suppose that the cost of one edge $\{u, v\}$ in $G$ is *decreased*; $\{u, v\}$ may or may not belong to $T$. Design an algorithm to either find a new MCST or to determine that $T$ is still an MCST. The more efficient your algorithm is, the more points you will be credited for this problem. Explain why your algorithm is correct and analyze its time complexity.