CMPSC 465 Data Structures & Algorithms
Fall 2025 Ke Chen and Yana Safonova **HW 8**

HW 8

Due November 03, 10:00 pm

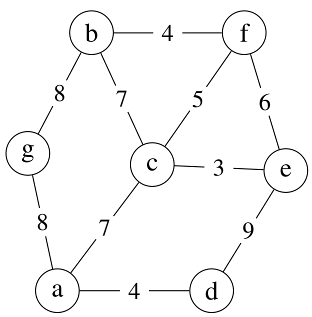
Instructions: You are encouraged to solve the problem sets on your own, or in groups of three to five people, but you must write your solutions strictly by yourself. You must explicitly acknowledge in your write-up all your collaborators, as well as any books, papers, web pages, etc. you got ideas from.

Formatting: Each part of each problem should begin on a new page. Each page should be clearly labeled with the problem number and the problem part. The pages of your homework submissions must be in order. **When submitting in Gradescope, make sure that you assign pages to problems from the rubric. You risk receiving no credit for it if you do not adhere to these guidelines.**

Late homework will not be accepted. Please, do not ask for extensions since we will provide solutions shortly after the due date. Remember that we will drop your lowest three scores.

This homework is due Monday, November 03, at 10:00 pm electronically. You need to submit it via Gradescope. Please ask on Campuswire about any details concerning Gradescope.

1. (20 pts.) **Find MST.** Consider the following graph:



(a) Run Kruskal's algorithm on the graph given below: give the order of edges that are added to the MST (whenever you have a choice, always choose the smallest edge in *lexicographic* order).

(b) Run Prim's algorithm on the graph given below: give the order of vertices that are added to the MST (whenever you have a choice, always choose the smallest vertex in *lexicographic* order).

2. (20 pts.) **Max Subgraph.** Let $G = (V, E)$ be an undirected graph. A graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. The *degree* of a node in a graph is the number of edges incident to it. Give an algorithm to find a largest subgraph (i.e. a subgraph with as many vertices as possible) G' of G such that every node in G' has degree (in G') at least k. Your algorithm should run in time $O(|V| + |E|)$.

3. (20 pts.) **Strange MST.** Let $G = (V, E)$ be a connected, undirected graph with edge weights $w : E \to \mathbb{R}$. Each edge $e \in E$ is labeled as either *mandatory*, *forbidden*, or *optional*. We wish to determine whether there exists a spanning tree $T \subseteq E$ that includes all mandatory edges, excludes all forbidden edges, and connects all vertices without forming a cycle.

(a) Describe how the Disjoint Set Union (DSU) structure can be used to check whether *mandatory* edges form any cycle or not.

(b) Modify Kruskal's algorithm to construct the minimum-cost feasible spanning tree if it exists, and analyze the correctness and runtime.

4. (20 pts.) **Linear Transformation.** Let $G = (V, E)$ be a connected, undirected graph with edge weights $w : E \to \mathbb{R}_{>0}$, and let T be a minimum spanning tree of G. Suppose each edge weight $w(e)$ is modified according to a linear function

$$w'(e) = a \cdot w(e) + b$$

for constants $a > 0$ and $b \geq 0$.

(a) Prove or disprove: T remains a minimum spanning tree of G under the new weights w'.

(b) Generalize your argument to the case where $a > 0$ but b can be any real number (possibly negative).

5. (20 pts.) **Dynamic MST.** Let $G = (V, E)$ be a connected, undirected graph with edge weights $w : E \to \mathbb{R}$, and let T be a minimum spanning tree of G. Assume all edge weights are distinct.

(a) Suppose a new edge $e' = (u, v)$ with weight $w'(e')$ is added to G. Describe an efficient method to decide whether T remains the MST.

(b) Suppose the weight of an existing edge e is increased from $w(e)$ to $w'(e)$. How will the MST change?