Homework 4 - Part b

Problem 1. (20 points) [Outputting a Particular MST]

Consider the scenario where we have a weighted graph $G = (V, E)$ where the weights of the edges are not necessarily unique. Kruskal's algorithm can return different spanning trees for the same input graph G, depending on how it breaks ties in edge costs when the edges are sorted. Prove that for each minimum spanning tree T of G, there is a way to sort the edges of G in Kruskal's algorithm so that the algorithm returns T.

Problem 2. (30 points) [Single-Vertex Degree Constrained MST]

Let $G = (V, E)$ be an undirected, connected graph where each edge $e \in E$ has a positive weight $w(e)$. We are given a specific vertex $v \in V$ and a positive integer maximum degree $r \geq 1$. The goal is to find a Spanning Tree T of G that minimizes the total weight $\sum_{e \in T} w(e)$, subject to the constraint that the degree of v in T, denoted $\deg_T(v)$, must be at most r.

Design an efficient algorithm that receives the graph G (as an adjacency list), the vertex v, and the constraint r, and outputs the minimum spanning tree subject to the constraint $\deg_T(v) \leq r$. If no spanning tree satisfies the degree constraint (e.g., G is a degree s-star and $r < s$ for the center vertex), state that it is "Impossible".

Hint: The degree constraint $\deg_T(v) \leq r$ prevents direct application of standard MST algorithms. A constrained MST must use $k \leq r$ of the edges incident to v. To find the optimal set of k edges, consider introducing a positive penalty λ to the weight of every edge incident to v:

$$w'(e) = \begin{cases} w(e) + \lambda & \text{if } v \in e \\ w(e) & \text{if } v \notin e \end{cases}$$

Observe how Kruskal's algorithm behaves with these modified weights.