Homework Assignment #6
Due: March 6, 2023 at 7:00 p.m.

1. We can model a computer network as a connected, undirected graph. Nodes represent computers and edges represent wires connecting pairs of computers. Each edge has an associated bandwidth (the number of bits per second that can be sent across the edge). The bandwidth of a path is the minimum bandwidth of any edge in the path.

Suppose we wish to select a spanning tree of the graph to use to stream a video from one source node s to all other nodes in the network. We want to ensure that the bandwidth of the path connecting s to each other node is as large as possible.

[3] **(a)** Given a graph G representing the network, modify Kruskal's algorithm to solve this problem. Your algorithm should be designed so that you can complete its proof of correctness as outlined in part (b).

[6] **(b)** Prove the following claim by filling in the outline provided.

Claim: The spanning tree T found by the algorithm in part (a) has the property that for every node $t \neq s$, the bandwidth of the path from s to t in T is greater than or equal to the bandwidth of every other path from s to t in G.

Proof Outline: Let t be any node in G other than s. Let B be the maximum bandwidth of any path from s to t in G. To derive a contradiction, suppose that B is greater than the bandwidth of the path from s to t in T. Let p be a path from s to t in G that has bandwidth B and contains the *smallest* possible number of non-T edges. (I.e., among all paths from s to t in G with bandwidth B, none of them contain fewer non-T edges than p does.)

Let (u, v) be some edge in p that is not in T. Let p_1 be the part of p before edge (u, v) and let p_2 be the part of the path after (u, v). Thus, $p = p_1 \cdot (u, v) \cdot p_2$.

(i) Explain why the edge (u, v) exists. I.e., explain why p must contain a non-T edge.

(ii) Let p_3 be the path from u to v in T. Argue that the bandwidth of every edge in p_3 is greater than or equal to the bandwidth of (u, v).

(iii) Describe how to construct a path from s to t in G that has bandwidth B and has *fewer* non-T edges than p. Remark: the path you construct does not have to be a simple path.

(iv) Explain why part (iii) completes the proof of the claim.

[1] **(c)** What is the running time of your algorithm on a graph with n nodes and m edges? Express your answer using Θ notation, as a function of n and m.