# Week 14: Lab

Module: Graphs

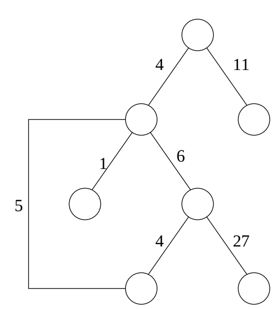
1. Suppose $G = (V, E)$ is a connected, undirected graph with more than $|V| - 1$ edges, and assume no two edges are the same. Is the heaviest edge in $G$ guaranteed to not be in the MST? If yes, justify. If no, give a counterexample.

2. Let G be a weighted, connected, undirected graph, and let e be the edge with lowest cost. Suppose that no other edge has the same or lower cost as e (so the lowest-cost edge is unique). Are we guaranteed that every minimum spanning tree of G will contain e? Why or why not?

3. Suppose G is an undirected graph, and suppose we want to compute a **maximum-weight** spanning tree (a ST of maximum overall weight). Describe how you would do it and include a brief justification of correctness.

4. Is the path between a pair of nodes $u$ and $v$ in a minimum spanning tree (i.e. the path composed of edges in the MST) necessarily the minimum weight path between $u$ and $v$ if we consider all the edges in the graph $G$? If yes, prove it. If no, provide a counterexample.

5. Consider an undirected weighted graph which is formed by taking a binary tree and adding an edge from *exactly one* of the leaves to another node in the tree. We call such a graph a *loop-tree*. An example of a loop-tree could be the following:




Let $n$ be the number of vertices in a loop-tree and assume that the graph is given in the normal edge-list representation without any extra information. In particular, the representation does not contain information about which vertex is the root.

(a) How many edges are in a graph of $n$ vertices?

(b) How long time would it take Prim's or Kruskal's algorithms to find the minimal spanning tree of a loop-tree?

(c) Describe and analyze a more efficient algorithm for finding the minimal spanning tree of a loop-tree. Remember to argue that the algorithm is correct.