CP 312, Fall 2017
Assignment 5 (8% of the final grade)
(due Monday, December 3, at 10:30 PM)

There are 6 questions in this assignment.

1. [5 marks] Give an example of a connected, weighted, undirected graph and a start vertex such that neither the BFS-tree nor the DFS-tree is an MST, regardless of how the adjacency lists are ordered. Argue that your graph and start vertex satisfy the above requirements.

2. [7 marks] A directed acyclic graph (DAG) is called a *lattice* if it has one vertex reachable from any other vertex (sink), and one vertex such that every other vertex can be reached from it (source). Give an efficient algorithm to determine if a given DAG is a lattice. Provide justification of the correctness and analyze running time complexity of your algorithm. Your algorithm has to have a running time in $O(|V| + |E|)$.

3. [6 marks] Prove that if the BFS and DFS spanning trees of a connected undirected graph G from the same start vertex s are equal to each other, then G is a tree.

4. [8 marks] The *distance* between two vertices in an undirected graph is the length of the shortest path between them. The *diameter* of an undirected graph is the maximum distance between vertices in the graph. Give an efficient algorithm to compute the diameter of a graph. Provide justification of the correctness and a its running time complexity.

5. [14 marks] Prove or disprove each of the following statements, where in each case $G = (V, E)$ is a connected undirected weighted graph with n vertices and m edges.

   (a) If G has $m \geq n$ edges and a unique heaviest edge e, then e is not part of any minimum spanning tree of G.

   (b) If G has $m \geq n$ edges and a unique lightest edge e, then e is part of every minimum spanning tree of G.

   (c) If e is a maximal weight edge of a cycle of G, then there is a minimum spanning tree of G that does not include e.

   (d) Prim's algorithm returns a minimum spanning tree of G even when the edge weights can be either positive or negative.

6. [10 marks] Given a directed graph with positive edge lengths and a specified vertex v in the graph, the "all-pairs v-constrained shortest path problem" is the problem of computing for each pair of vertices i and j the shortest path from i to j that goes through the vertex v. If no such path exists, the answer is ∞. Describe an algorithm that takes a graph $G = (V, E)$ and vertex v as input parameters and computes values $L(i, j)$ that represent the length of v-constrained shortest path from i to j for all $1 \leq i, j \leq |V|$, $i \neq v$, $j \neq v$. Prove your algorithm correct. Your algorithm should have a running time in $O(|V|^2)$.