Homework Assignment #9
Due: March 27, 2020 at 2:30 p.m.

Note: the answers to the questions on this assignment can be very short.

1. Suppose you have a directed graph with n nodes and $\Theta(n)$ edges. Assume the graph is given using an adjacency list representation. Each edge has a non-negative edge weight. There is one special vertex w in the graph. You wish to find, for every pair of nodes u and v, the shortest path from u to v that passes through w.

(a) Describe an algorithm to solve this problem. Your algorithm should be as efficient as possible. You can use any algorithm discussed in class or the textbook as a subroutine. Describe how any data structures used in your algorithm are implemented.

(b) What is the worst-case running time of your algorithm. Express your answer in terms of n using Θ notation.

2. Consider an undirected graph with non-negative edge weights. The graph has n nodes and m edges and assume it is given using an adjacency list representation. As usual, $distance(u, v)$ is the length of the shortest path from u to v. Let $D = \max_{u,v} distance(u, v)$ be the maximum distance between any two nodes in the graph.

(a) How can you compute D in worst-case $\Theta(n^3)$ time?

(b) Suppose we just want to get an approximate value of D. Let s be any node in the graph. Let $D' = \max_{v} distance(s, v)$ be the maximum distance from s to any other node in the graph. Show that the value of D' is within a factor of 2 of D and can be computed much faster than the exact computation in part (a).