This assignment is due Friday Oct 12th at 6pm. Enjoy your Fall break!

Problem 1. [20 points]
Some friends of yours work on wireless networks, and they’re currently studying the properties of a network of n mobile devices. As the devices move around (actually, as their human owners move around), they define a graph at any point in time as follows: there is a node representing each of the n devices, and there is an edge between device i and device j if the physical locations of i and j are no more than 500 meters apart. (If so, we say that i and j are “in range” of each other.)

They’d like it to be the case that the network of devices is connected at all times, and so they’ve constrained the motion of the devices to satisfy the following property: at all times, each device i is within 500 meters of at least n/2 of the other devices. (We’ll assume n is an even number.)

What they’d like to know is: Does this property by itself guarantee that the network will remain connected? Here’s a concrete way to formulate the question as a claim about graphs.

Claim: Let G be a graph on n nodes, where n is an even number. If every node of G has degree at least n/2, then G is connected.

(a) Decide whether you think the claim is true or false.
(b) Give a counter-example if it is false, or write a clear proof on why it is true.

Problem 2. [20 points - Proof Problem]
There’s a natural intuition that two nodes that are far apart in a communication network—separated by many hops—have a more tenuous connection than two nodes that are close together. There are a number of algorithmic results that are based to some extent on different ways of making this notion precise. Here’s one that involves the susceptibility of paths to the deletion of nodes.

Suppose that an n-node undirected graph G = (V, E) contains two nodes s and t such that the distance between s and t is strictly greater than n/2.

(a) [Proof Problem, submit separately] Prove that there must exist some node v, not equal to either s or t, such that deleting v from G destroys all s − t paths. (In other words, explain that the graph obtained from G by deleting v contains no path from s to t.)

(b) Give an algorithm with running time O(m + n) to find such a node v.

Problem 3. [20 points]
The algorithm described in Section 3.6 for computing a topological ordering of a DAG repeatedly finds a node with no incoming edges and deletes it. This will eventually produce a topological ordering, provided that the input graph really is a DAG.

But suppose that we’re given an arbitrary graph that may or may not be a DAG. Extend the topological ordering algorithm so that, given an input directed graph G, it outputs one of two things:

(a) a topological ordering, thus establishing that G is a DAG; (b) a cycle in G, thus establishing that G is not a DAG.

The running time of your algorithm should be O(m + n) for a directed graph with n nodes and m edges.