Assignment 3

Due: Wed, Oct 17, 2018 @ 11.59pm

Directions:

- Your solutions must be typeset. LaTeX is recommended.
- You must upload your solutions as a PDF file on Canvas before the deadline.
- You don't have to include your solutions to the programming problems in the PDF file.

Problem 1

Give an algorithm to check whether a given undirected graph $G = (V, E)$ is a *t*ree. What is the running time of your algorithm?

Problem 2

Show that, for any DAG, there is always a node u with no incoming edges and a node v with no outgoing edges

Problem 3

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.

Decide whether you think the claim is true or false. Give a proof if you think the claim is true, or give a counterexample otherwise.

Problem 4

A server has n customers waiting to be served. The service time required by each customer is known in advance: it is t_i minutes for customer i. So if, for example, the customers are served in order of increasing i, then the i-th customer has to wait $\sum_{j=1}^{i} t_j$ minutes.

We wish to minimize the total waiting time

$$T = \sum_{i=1}^{n} (\text{time spent waiting by customer } i)$$

Give an efficient algorithm for computing the optimal order in which to process the customers.

Problem 5

Alice wants to throw a party and is deciding whom to call. She has n people to choose from, and she has made up a list of which pairs of these people know each other. She wants to pick *as many people as possible*, subject to a constraint: at the party, each person should have at least five other people whom they know.

(a) Give a high-level description of an efficient algorithm that takes as input the list of n people and the list of pairs who know each other and outputs the best choice of party invitees. Argue that this scheme is correct. (Hint: instead of thinking about who to select, think about who to NOT select)

(b) Give an efficient implementation of the scheme from part (a), and analyze its running time in terms of n.

Problem 6

Complete **at least 4 problems** from the following contest: `https://vjudge.net/contest/258852`.