CS41 Homework 4

This homework is due at 11:59PM on Sunday, October 4.. Write your solution using LaTeX. Submit this homework using **github** as a file called **hw4.tex**. This is a **partnered homework**. You should primarily be discussing problems with your homework partner.

It's ok to discuss approaches at a high level with others. However, you should not reveal specific details of a solution, nor should you show your written solution to anyone else. The only exception to this rule is work you've done with a lab partner *while in lab.* In this case, note (in your **homework submission poll**) who you've worked with and what parts were solved during lab.

1. **Paths in Graphs.** Recall that a path P in a graph $G = (V, E)$ is a sequence of vertices $P = [v_1, \ldots, v_k]$ such that for all $1 \leq i < k$ there is an edge $(v_i, v_{i+1}) \in E$. P is *simple* if all v_i's are distinct.

   In this problem, you will examine different graphs and consider how many different paths can exist in the graph.

   (a) Describe a graph G_1 on n vertices where between any two distinct vertices there are zero simple paths.

   (b) Describe a graph G_2 on n vertices where between any two distinct vertices there is exactly one simple path.

   (c) Describe a graph G_3 on n vertices where between any two distinct vertices there are exactly two simple paths.

   (d) Describe a graph $G_4 = (V, E)$ on n vertices and two distinct vertices $s, t \in V$ such that there are $2^{\Omega(n)}$ simple $s \leadsto t$ paths.

2. **Network Connectivity**(Kleinberg and Tardos, 3.7) 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 200 meters apart. (If so, 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 200 meters of at least $n/2$ of the other devices. (Assume that 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 1. *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, and give a proof of either the claim or its negation.

3. **Liars and Friars.** To escape from some recent bad events, you sail off to a tropical island. This island is populated by n inhabitants; each inhabitant is either a *liar* or a *friar*. A friar always tells the truth, but liars cannot be trusted. You want to find the best place to eat dinner, and you definitely do not want to ask a liar (they will recommend bagel bar at Sharples, or Starbucks), so you would like to identify at least one friar. To help identify a friar, you can pair up any two inhabitants A, B and ask each to identify the other. If either A or B identifies that the other is a liar, then at least one of A and B is a liar. If both claim the other is a friar, then either both are liars or both are friars.

   (a) Show that if more than $n/2$ inhabitants are liars, it is not generally possible to identify a friar. You may assume liars collaborate to convince you they are friars.

   (b) Now, suppose that more than $n/2$ people are friars. Show that with at most $\lfloor \frac{n}{2} \rfloor$ pairwise comparisons, it is possible to reduce the problem to one of nearly half the original size. **Hint:** the algorithmic goal of this part is to produce an algorithm that, given n inhabitants (over half of whom tell the truth) returns a list of $\approx n/2$ inhabitants, over half of whom tell the truth.

   (c) Use your solution to Part (3b) to construct an algorithm which identifies a single friar.

   (d) Show how to find all friars, assuming that more than half of the n inhabitants are friars.

   Note: Do not make any assumptions about n, e.g., do not assume n is odd, or n is a power of two, etc.

4. **(Extra Challenge).** For a positive integer k, call a graph k-colorable if the vertices can be properly colored using k colors. In other words, a bipartite graph is two-colorable. In this problem, you will investigate algorithms dealing with three-colorable graphs.

   - Design and analyze an algorithm which takes as input a graph $G = (V, E)$ and returns YES if G is three-colorable, and NO otherwise.
   - Design and analyze an efficient algorithm which takes as input a *three-colorable* graph $G = (V, E)$ and colors the vertices of the graph using $O(\sqrt{n})$ colors. (Note: while the input graph *is* three-colorable, it does not mean that we know what that coloring is!)