CS41 Homework 5

This homework is due at 11:59PM on Sunday, October 7. Write your solution using LaTeX. Submit this homework using **github** as a **.tex** file. 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 **README** file) who you've worked with and what parts were solved during lab.

All of the problems on this homework are on directed, unweighted graphs. Give low-level pseudocode for problem 1. For other problems, provide high-level pseudocde or an English description of the algorithm. For these problems, **focus on algorithm correctness**. Your algorithms should be efficient, but runtime analysis will not be a large factor in your grade this week.

1. **All-Pairs Shortest Paths.** Design and analyze a polynomial-time algorithm that takes a directed graph $G = (V, E)$ and for all $u, v \in V$ computes the length of the shortest $u \rightsquigarrow v$ path or determines that no such path exists.

2. **Computer virus proliferation (K&T 3.11).** You are helping some security analysts monitor a collection of networked computers, tracking the spread of a virus. There are n computers in the system, call them C_1, C_2, $\ldots$, C_n. You are given a trace indicating the times at which pairs of computers communicated. A trace consists of m triples (C_i, C_j, k) that indicate that C_i communicated with C_j at time t_k. At that time, the virus could have spread between C_i and C_j.

   We assume that the trace holds the triples in sorted order by time. For simplicity, assume that each pair of computers communicates at most once over the time of the trace. Also, it is possible to have pairs (C_s, C_j, k) and (C_t, C_j, k): this would indicate that computer C_j opened connections to both C_s and C_t at time t_k, allowing the virus to spread any way among the three machines.

   Design and analyze an efficient algorithm that, given as input a collection of time-sorted trace data and a virus query, answers the question "if the virus was introduced to C_i at time x, could it spread to C_j at time y?" That is, is there a sequence of communications that could have lead to the virus moving from C_i to C_j?

3. **Ethnographers (K&T 3.12).** You're helping a group of ethnographers analyze some oral history data they've collected by interviewing members of a village to learn about the lives of people who have lived there over the past two hundred years.

   From these interviews, they've learned about a set of n people (all now deceased), whom we'll denote $P_1, P_2, \ldots, P_n$. They've also collected facts about when these people lived relative to one another. Each fact has one of the following two forms:

   - for some i and j, person P_i died before person P_j was born; or
   - for some i and j, the lifespans of P_i and P_j overlapped at least partially.

Naturally, the ethnographers are not sure that all these facts are correct; memories are not very good, and a lot of this was passed down by word of mouth. So what they'd like you to determine is whether the data they've collected is at least *internally consistent*, in the sense that there could have existed a set of people for which all the facts they've learned simultaneously hold.

Give an efficient algorithm to do this: either it should propose dates of birth and death for each of the n people so that all the facts hold true, or it should report (correctly) that no such dates can exist—that is, the facts collected by the ethnographers are not internally consistent.

4. **Enemies on the Move.** Alice and Bob are very active students at University of Pennsylvania. They used to be best friends but now despise each other. Alice and Bob can't stand to be in the same room, or even nearby. However, they each take many classes and are active in several clubs. Is it even possible to avoid each other?

   This can be modeled as a graph problem. The input consists of:

   - a directed graph $G = (V, E)$,
   - an integer $k \leq n$,
   - start vertices s_A and $s_B \in V$, and
   - end vertices t_A and $t_B \in V$.

   In this problem, Alice starts at s_A and wants to travel to t_A, while Bob starts at s_B and wants to travel to t_B. At each time step, either Alice or Bob moves along a single edge. (You can assume they move separately.) At all times, Alice and Bob must be at least k edges apart.

   Design and analyze a polynomial-time algorithm that determines if Alice and Bob can get where they want to go while maintaining distance.

   (**Hint:** It might be helpful to use your solution to the all-pairs-shortest-paths problem from problem(1) as a subroutine.)