CS41 Homework 6

This homework is due at 11:59PM on **Friday, October 12**. **Note the unusual deadline.** This is a **6 point** homework. 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.

In this assignment, you will write some code and some LaTeX. The provided Makefile should be able to compile both. Type `make all` to compile both your C++ program and your homework assignment.

1. **Implementing Cycle Detection for Directed Graphs.** For this problem, your task is to write a program that takes as input a directed graph and determines whether the input graph contains a cycle.

   - You may write your program in C++ or in Python.
   - If you write in C++, we've given you most of the code as a starting point, using the graph implementations you might have seen in CS35.
   - We've also given you a program file `detectCycle.cpp` which takes as input the name of the file containing your input graph and prints out whether this graph has a cycle. **You should only need to implement the `detectCycle` function in `graphAlgorithms.cpp`.**
   - The file format for the directed graphs is as follows. The first line of the file contains a single integer n. The next n lines of code each have a single string giving the name of a vertex. The line after that contains a single integer m. The last m lines of the file each contain two strings $src, dest$ representing a directed edge from src to $dest$.

2. **Changing edge costs** (Kleinberg and Tardos, 4.2)
   For each of the following two statements, decide whether it is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

   (a) Suppose we are given an instance of the Minimum Spanning Tree (MST) problem on a graph $G = (V, E)$, with edge costs that are all positive and distinct. Let T be a minimum spanning tree for this instance. Now, suppose we replace each cost c_e by its square c_e^2, thereby creating a new instance of the problem with the same graph but different costs.

   TRUE or FALSE? T must still be a minimum spanning tree for this new instance.

   (b) Suppose we are given an instance of the Shortest $s \rightsquigarrow t$ Path Problem on a directed graph $G = (V, E)$, with edge costs that are all positive and distinct. Let P be a a minimum-cost $s \rightsquigarrow t$ path for this instance. Now, suppose we replace each edge cost c_e by its square c_e^2, thereby creating a new instance of the problem with the same graph but different costs.

   TRUE or FALSE? P must still be a minimum $s \rightsquigarrow t$ path for this new instance.

3. **(Kleinberg and Tardos, 4.6)** Your friend is working as a camp counselor, and he is in charge of organizing activities for a set of junior-high-school-age campers. One of his plans is the following mini-triathalon exercise: each contestant must swim 20 laps of a pool, then bike 10 miles, then run 3 miles. The plan is to send the contestants out in a staggered fashion, via the following rule: the contestants must use the pool one at a time. In other words, first one contestant swims the 20 laps, gets out, and starts biking. As soon as this person is out of the pool, a second contestant begins swimming the 20 laps; as soon as he or she is out and starts biking, a third contestant begins swimming...and so on.)

Each contestant has a projected *swimming time* (the expected time it will take him or her to complete the 20 laps) a projected *biking time* (the expected time it will take him or her to complete the 10 miles of bicycling), and a projected running time (the time it will take him or her to complete the 3 miles of running). Your friend wants to decide on a *schedule* for the triathalon: an order in which to sequence the starts of the contestants. Let's say that the *completion time* of a schedule is the earliest time at which all contestants will be finished with all three legs of the triathalon, assuming they each spend exactly their projected swimming, biking, and running times on the three parts. (Again, note that participants can bike and run simultaneously, but at most one person can be in the pool at any time.) What's the best order for sending people out, if one wants the whole competition to be over as early as possible? More precisely, give an efficient algorithm that produces a schedule whose completion time is as small as possible.