CS 231: Fundamentals of Algorithms **Fall 2021**

Assignment #3

Your Name:

Collaborators:

Total time spent:

Problem 1. [10 points]
You're helping a group of ethnographers analyze some oral history data they have 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 have learned about a set of n people (all of them now deceased), whom we'll denote as $P_1, P_2, ..., P_n$. They have 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 life spans of P_i and P_j overlapped at least partially.

Naturally, they are not sure that all these facts are correct; memories are not so 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 that can verify the information collected by these ethnographers. In other words, it should report whether the events recorded by them are consistent or not.

Your answer should clearly describe the algorithm (you can write it in plain English sentences), the input given to the algorithm, and the output expected.

Problem 2. [5 points]
The basic question is: Can a man or a woman end up better off by lying about his or her preferences? More concretely, we suppose each participant has a true preference order. Now consider a woman w. Suppose w prefers man m to m', but both m and m' are low on her list of preferences. Can it be the case that by switching the order of m and m' on her list of preferences (i.e., by falsely claiming that she prefers m' to m) and running the algorithm with this false preference list, w will end up with a man m'' that she truly prefers to both m and m'? (We can ask the same question for men, but will focus on the case of women for purposes of this question.)

Resolve this question by doing one of the following two things:
(a) Give a proof that, for any set of preference lists, switching the order of a pair on the list cannot improve a woman's partner in the Gale- Shapley algorithm; or
(b) Give an example of a set of preference lists for which there is a switch that would improve the partner of a woman who switched preferences.

Problem 3. [5 points] - Proof problem to be submitted separately
We have a connected graph $G = (V, E)$, and a specific vertex $u \in V$. Suppose we compute a depth-first search tree rooted at u, and obtain a tree T that includes all nodes of G. Suppose we then compute a breadth-first search tree rooted at u, and obtain the same tree T. Prove that $G = T$. (**Hint:** if T is both a depth-first search tree and a breadth-first search tree rooted at u, then G cannot contain any edges that do not belong to T.)

Problem 4. [5 points]
Let's consider a long, quiet country road with house scattered very sparsely along it. (We can picture the road as a long line segment, with an eastern endpoint and a western endpoint.) Further, let's suppose that

despite the bucolic setting, the residents of all these houses are avid cell phone users. You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations.

(a) Give an efficient algorithm that achieves this goal, using the minimum number of base stations.
(b) Prove that your algorithm is correct.

Problem 5. [10 points]
You are consulting for a trucking company that does a large amount of business shipping packages between New York and Boston. The volume is high enough that they have to send a number of trucks each day between the two locations. Trucks have a fixed limit W on the maximum amount of weight they are allowed to carry. Boxes arrive at the New York station one by one, and each package i has a weight w_i. The trucking station is quite small, so at most one truck can be at the station at any time.

Company policy requires that boxes are shipped in the order they arrive; otherwise, a customer might get upset upon seeing a box that arrived after his make it to Boston faster. At the moment, the company is using a simple greedy algorithm for packing: they pack boxes in the order they arrive, and whenever the next box does not fit, they send the truck on its way.

But they wonder if they might be using too many trucks, and they want your opinion on whether the situation can be improved. Here is how they are thinking. Maybe one could decrease the number of trucks needed by sometimes sending off a truck that was less full, and in this way allow the next few trucks to be better packed.

(a) From your understanding of the algorithms presented in this problem. Rewrite the optimal greedy algorithm proposed in this problem (pseudocode is recommended).
(b) To establish the optimality of this greedy packing algorithm, you need to identify a measure under which it "stays ahead" of all other solutions. What is that measure?
(c) Prove that, for a given set of boxes with specified weights, the greedy algorithm currently in use actually minimizes the number of trucks that are needed. Your proof should follow the type of analysis we used for the Interval Scheduling Problem.

Problem 6. [10 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; or
(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.

Problem 7. [5 points]
Use mathematical induction to show that when n is an exact power of 2, the solution of the recurrence

$$T(n) = \begin{cases} 2, & \text{if} n = 2 \\ 2\ \mathrm{T(n/2)} + \mathrm{n}, & \text{if} n = 2^k, k > 1 \end{cases}$$

is T(n) = n lg n. (Hint: Your induction should proceed on k, not on n).