CS7800: Advanced Algorithms. Fall 2017
Homework 4

Instructor: Jonathan Ullman   TA: Albert Cheu

Due Friday, October 6th at 11:59pm
(Email to neu.cs7800@gmail.com)

Homework Guidelines

Collaboration Policy. Collaboration on homework problems is permitted, however it will serve you well on the exams if you solve the problems by yourself. If you choose to collaborate, you may discuss the homework with at most 2 other students currently enrolled in the class. Finding answers to problems on the Web or from other outside sources (these include anyone not enrolled in the class) is strictly forbidden.

You must write up each problem solution by yourself without assistance, even if you collaborate with others to solve the problem. You must also identify your collaborators. If you did not work with anyone, indicate that on your submission. If asked, you must be able to explain your solution to the instructors.

Preparing and Submitting Solutions. You must type your solutions using LaTeX. Please use an 11-pt or larger font. Please submit both the source and PDF files using the naming conventions `lastname_hw4.tex` and `lastname_hw4.pdf`. Your name must be on the first page of the PDF.

Solution guidelines For problems that require you to provide an algorithm, you must give the following: (1) a precise description of the algorithm in English and, if helpful, pseudocode, (2) a proof of correctness, (3) an analysis of running time. You may use any facts from class in your analysis and you may use any algorithms from class as subroutines in your solution.

You should be as clear and concise as possible in your write-up of solutions. Communication of technical material is an important skill, so clarity is as important as correctness. A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error and because it is easier to read and understand. Points might be subtracted for solutions that are too long.

Problem 1 (Longest Path, 10 pts)**.**

An ordered graph $G = (V, E)$ is a directed graph with nodes $v_1, \ldots, v_n$ and two properties: (1) all edges are of the form (v_i, v_j) for $j \geq i$ and (2) for every $i = 1, \ldots, n-1$ there is at least one edge (v_i, v_j) leaving v_i. Design a polynomial time algorithm that takes as input an ordered graph G and outputs a *longest* path from v_1 to v_n. Clearly describe your algorithm, prove that it is correct, and analyze its running time and space usage.

Problem 2 (Staffing on a Budget, 10 pts)**.**

You are running a *Magic: The Gathering* convention and you need to find people to staff the registration desk. Unfortunately, *Magic: The Gathering* isn't as popular as it once was, so you'll need to do it on a shoestring budget. There are n potential staffers $\{1, \ldots, n\}$. Each potential staffers i has given you a time window in which they can work, represented by an interval $[s_i, t_i]$, and a price for their services, represented by a number p_i. Let $[s, t]$ represent the time window for the convention. A set of staffers $S \subseteq \{1, \ldots, n\}$ is a *complete set* if $\bigcup_{i \in S} [s_i, t_i] \supseteq [s, t]$ and its total cost is $\sum_{i \in S} p_i$.

Design a polynomial time algorithm that takes a time window $[s, t]$ for the convention and time-windows and prices $([s_i, t_i], p_i)$ for n staffers and outputs a complete set with minimum minimum total cost. For simplicity you may assume that at every time in $[s, t]$, some staffer can work that time, and that all values $s, t, s_1, t_1, \ldots, s_n, t_n$ are distinct. Clearly describe your algorithm, prove it is correct, and analyze its running time and space usage. Don't forget to output the set of staffers, not just the total cost.

(**Bonus:** Find an algorithm running in $O(n \log n)$ time.)