COMPSCI330 Design and Analysis of Algorithms
Assignment 8

Due Date: Wednesday, April 22, 2020

Guidelines

- **Describing Algorithms** If you are asked to provide an algorithm, you should clearly define each step of the procedure, establish its correctness, and then analyze its overall running time. There is no need to write pseudo-code; an unambiguous description of your algorithm in plain text will suffice. If the running time of your algorithm is worse than the suggested running time, you might receive partial credits.

- **Typesetting and Submission** Please submit the problems to GradeScope. You will be asked to **label your solution for individual problems**. Failing to label your solution can cost you 5% of the total points (3 points out of 60 for this homework).

- LaTeX is preferred, but answers typed with other software and converted to pdf is also accepted. Please make sure you submit to the correct problem, and your file can be opened by standard pdf reader. **Handwritten answers or pdf files that cannot be opened will not be graded. (Exceptions apply to students who don't have regular access to computers.)**

- **Timing** Please start early. The problems are difficult and they can take hours to solve. The time you spend on finding the proof can be much longer than the time to write. If you need more time for your homework please use this form and submit a STINF.

- **Collaboration Policy** Please check this page for the collaboration policy. You are **not** allowed to discuss homework problems in groups of more than 3 students. **Failure to adhere to these guidelines will be promptly reported to the relevant authority without exception.**

Special Policy for Last Homework

As this is commonly considered to be the most difficult problem set in the course, in this difficult time we are providing an option to (effectively) skip one question.

To take advantage of this, when you submit your homework, you need to clearly state which problem you are skipping. You can do that by just write “I am skipping this problem.” in place of the solution for that problem. You will automatically receive 20 points for that problem (note that this also applies to P2 which has two different subproblems, you will just receive full credit for both (a) and (b)). To qualify for the skipping option, you need to attempt at least one other problem (i.e., a completely blank submission will not receive 20 points).

If you have time, we still encourage you to work on all three problems. In that case you might want to receive feedback while still making sure you get 20 points on a certain problem (for example, maybe you are uncertain about your solution to Problem 3). In that case, you should just write “I am skipping this problem but I would still like feedback.” Then write your solution after that. The graders will grade your solution, but will add a +20 so your final score will be 20 for that problem. You will still be able to see the regular feedback on gradescope.

Problem 1 (STORE PLANNING)**.** (20 points) SET COVER is a classical NP-complete problem. In this problem, there are n sets $S_1, S_2, ..., S_n$. All of these n sets are subsets of $\{1, 2, ..., m\}$. The input of the problem is $n, m, k(1 \leq k \leq n)$ and a list of elements for each set. The answer to SET COVER is YES, if and only if there is a way to select k out of the n sets $(S_{i_1}, S_{i_2}, ..., S_{i_k})$ such that the union of these sets covers every element in $\{1, 2, ..., m\}$. That is, $\cup_{j=1}^{k} S_{i_j} = \{1, 2, ..., m\}$. You can assume that every element appears in at least one set.

STORE PLANNING is a different problem. In this problem, there are m customers and n potential store locations. Store location i can serve a subset of customers $S_i \subset \{1, 2, ..., m\}$. Each customer needs to be served by at least one store. Opening store i costs $X + |S_i|$ (where $|S_i|$ is the size of set S_i). The input to the STORE PLANNING problem consists of $n, m, S_i's(i = 1, 2, ..., n), X$ and a value K, the answer is YES if there is a way to serve all customers with total cost less than K. You can assume that every customer can be served by at least one potential store.

Based on the fact that SET COVER is NP-complete, prove STORE PLANNING is NP-hard.

Hint: What happens if you set X to be large?

Problem 2 (MIN DIRECTED GRAPH)**.** (20 points) Recall the minimum spanning tree problem tries to find an undirected graph with minimum total edge weights such that all vertices are connected. What if the graph is directed? Given a directed graph G with n vertices and m edges and a number k, the MIN DIRECTED GRAPH problem wants to decide whether there exists a subset of k edges, such there is a path from any vertex u to any vertex v using only these edges.
(a) (5 points) Show that MIN DIRECTED GRAPH is in NP.
(b) (15 points) HAMILTONIAN CYCLE is a classical NP-complete problem. In this problem, given a directed graph with n vertices and m edges, the problem wants to decide whether there exists a cycle that visit all vertices exactly once. Based on the fact that HAMILTONIAN CYCLE is NP-complete, prove MIN DIRECTED GRAPH is NP-complete.

Problem 3 (SOCIAL DISTANCING)**.** (20 Points) City X is implementing a social distancing policy during this pandemic. Specifically, in a park which is described by an undirected graph G (with n vertices and m edges), no one is allowed to be on adjacent vertices at any given time. There are k people in the park, person i wants to go from vertex s_i to vertex t_i (these vertices are not adjacent to each other). At each step, a person can move along one edge in the graph, or stand still. The SOCIAL DISTANCING problem wants to decide whether there is a way for everyone to go from their respective starting point s_i to their ending point t_i within p steps, while still observing the social distancing policy.

INDEPENDENT SET is a classical NP-complete problem. In this problem, there is an undirected graph G with n vertices and m edges, the goal is to decide whether there exists k vertices such that no two vertices are connected by an edge.

Based on the fact that INDEPENDENT SET is NP-complete, prove SOCIAL DISTANCING is NP-hard.

Hint: 1. You can fix p in SOCIAL DISTANCING to the distance between s_i and t_i, so everyone needs to move along an edge in every step; 2. You might need to create addtional vertices for s_i and t_i, how many of them do you need?