University of Washington
Department of Computer Science and Engineering
CSE 417, Autumn 2020

December 3, 2020

Homework 9, Due Sunday, December 13, 2020

Problem 1 (10 Points):

(Kleinberg-Tardos, Based on exercise 9, Page 419) Network flow issues come up in dealing with natural disasters and other crises, since major unexpected events often require the movement and evacuation of large numbers of people in a short amount of time.

Consider the following scenario. Due to a virus outbreak in a region, paramedics have identified a set of n infected people distributed across the region who need to be rushed to hospitals. There are k hospitals in the region, and each of the n people needs to be brought to a hospital that is within a half-hour's driving time of their current location (so different people will have different options for hospitals, depending on where they are right now).

At the same time, one does not want to overload any one of the hospitals by sending too many patients its way. The paramedics are in touch by cell phone, and they want to collectively work out whether they can choose a hospital for each of the sick people in such a way that the load on the hospitals is balanced: Each hospital receives at most $\lceil n/k \rceil$ people.

Give a polynomial-time algorithm that takes the given information about the people's locations and determines whether this is possible.

Problem 2 (10 points) :

A group of traders are leaving India and need to convert their Rupees into various international currencies. There are n traders and m currencies. Trader i has T_i Rupees to convert. The bank has B_j Rupees worth of currency j. Trader i is willing to trade up to C_{ij} Rupees for currency j. (For example, a trader with 1000 rupees might be willing to convert up to 700 Rupees for USD, up to 500 Rupees for Japaneses Yen, and up to 500 Rupees for Euros). Assuming that all traders give their requests to the bank at the same time, describe an algorithm that the bank can use to satisfy the requests (if it can).

Problem 3 (10 points):

Answer the following questions with "yes", "no", or "unknown, as this would resolve the P vs. NP question." Give a brief explanation of your answer.

Define the decision version of the Interval Scheduling Problem as follows: Given a collection of intervals on a time-line, and a bound k, does the collection contain a subset of nonoverlapping intervals of size at least k?

a) Question: Is it the case that Interval Scheduling $\leq_P$ Vertex Cover?

b) Question: Is it the case that Independent Set $\leq_P$ Interval Scheduling?

Problem 4 (10 points):

Suppose that you have an $O(n^3)$ time algorithm for the Hamiltonian Circuit Problem. Prove that $P = NP$.

Problem 5 (10 points):

(Kleinberg-Tardos, Page 507, Problem 7). Since the 3-Dimensional Matching Problem is NP-complete, it is natural to expect that the corresponding 4-Dimensional Matching Problem is at least as hard. Let us define *4-Dimensional Matching* as follows. Given sets W, X, Y, and Z, each of size n, and a collection C of ordered 4-tuples of the form (w_i, x_j, y_k, z_l), do there exist n 4-tuples from C so that no two have an element in common?
Prove that 4-Dimensional Matching is NP-Complete.

Problem 6 (10 Points):

(Kleinberg-Tardos, Page 513, Problem 17). You are given a directed graph $G = (V, E)$ with weights w_e on its edges $e \in E$. The weights can be negative or positive. The *Zero-Weight-Cycle Problem* is to decide if there is a simple cycle in G so that the sum of the edge weights on this cycle is exactly 0. Prove that the Zero-Weight-Cycle problem is NP-Complete. (Hint: Hamiltonian PATH)