Foundations of Algorithms, Spring 2018: Homework 6

Due: Wednesday, April 25, 11:59pm

Problem 1 (20 points - 12 for implementation, 8 for written response)

The evil wizard has placed you, the good wizard, in a dungeon maze. You must safely reach the exit before the wizard casts a spell that turns you into a paperclip. This particular maze may have many paths to reach the exit safely, but there is a twist. Various barricades have been placed throughout the maze. You have a limited number of magic vials. Each vial can be used to eliminate one barricade.

Let n represent the number of points of interest in the maze. This number includes your entry point, the exit point, and each of the barricade points. Let m represent the number of connections in the maze. They take you from one point of interest to another point of interest. Each connection has a time associated with it that indicates how long it takes you to travel between those two points of interest.

Design an $O(mn)$ algorithm that determines whether it is possible for you to safely reach the exit before the evil wizard turns you into a paperclip.

Problem 2 (10 points)

Recall that the Edmonds-Karp algorithm refines the idea of Ford-Fulkerson in the following way: in every iteration, the algorithm chooses the augmenting path that uses the fewest edges (if there are multiple such paths, it chooses one arbitrarily). Find a graph for which in some iteration the Edmonds-Karp algorithm has to choose a path that uses a backward edge. Run the algorithm (by hand) on your graph - more precisely, for every iteration draw the residual graph and show the augmenting path taken by the algorithm as well as the flow after adding the augmenting path.

Problem 3 (20 points: 12 for implementation / 8 for writeup)

Consider the problem of students registering for courses. For each student, you are given a set of courses that the student is willing and able to take. No student may take more than 3 courses. Each course also has a maximum number of students than can enroll, though this may be different for different courses. Create an efficient algorithm that will compute the maximum sum of registered students over all courses.

Problem 4 (10 points)

The Traveling Salesman Problem (TSP) is defined as follows: given a complete weighted undirected graph on n vertices (i.e. there is an edge between every pair of vertices) and a number $k > 0$, does there exist a cycle going through every vertex exactly once with total weight at most k? (The weight of a cycle is the sum of the weights of the edges forming the cycle.)

The Hamiltonian Cycle (HC) problem is defined as follows: given an undirected graph, does there exist a cycle that goes through every vertex exactly once?

Show that HC is polynomially-reducible to TSP, i.e. HC $\leq_P$ TSP. In other words, assume that we have a black box that solves TSP (the input to the block box is n, the weights of all edges, and the number k). We need to:

(a) Let G be an input of HC. Transform it into an input for the black box.

(b) After the black box produces an answer (YES or NO), transform it into the answer to HC with input G.

Your solution should describe both steps (a) and (b) and argue why the construction works. The transformations in steps (a) and (b) need to be done in polynomial time.

Note: both TSP and HC problems are known to be NP-complete. This means that we do not know if they can be solved in polynomial time but most people think that no polynomial-time algorithm exists. Nevertheless, we can pretend to have a black box for TSP and see if such a black box would help solving HC (and other problems).