Foundations of Algorithms, Spring 2021: Homework 6

Due: Friday, April 30, 11:59pm

Because this is the last homework, it includes material we are just now covering in class. I am extending the due date by two additional days to make sure you have time to answer the question on the topic of complexity.

Note that there are only two problems requiring an implementation. I recommend you complete problem 1 first, which is one of the implementation questions, and whose concepts we have covered completely.

Problem 1 (20 points: 14 for implementation / 6 for writeup)

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 (20 points: 14 for implementation / 6 for writeup)

Suppose you are running a daycare center and you need to get the children dressed to take them outside for a walk. Every child needs a hat, a pair of mittens, and a winter jacket. The trouble is that all the clothes got mixed up and the children do not remember (or do not want to tell you) which clothes are whose. Every child has certain preferences for which clothes they are willing to put on. For example, Izzy is ok with the green or the red hat, the wool mittens, and either the long blue jacket, or the purple jacket, or the jacket with yellow stripes. You record the preferences for all the children and now you are trying to figure out if you can get every child dressed properly for the walk.

Formally, suppose there are n children, a hats, b pairs of mittens, and c jackets (there might be extra or missing clothing items so it might happen that we do not have exactly n hats, mittens, or jackets). For every child we have a list of acceptable hats, mittens, and jackets. Design an $O(n^2 \max\{a, b, c\})$ algorithm that determines whether it is possible to get every child dressed with a hat, a pair of mittens, and a winter jacket that the child finds acceptable. Of course, no two children can share the same item of clothing.

Problem 3 (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 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 black box is n, the weights of all the edges, and the number k; the output is YES if there exists such cycle and NO otherwise). 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 black box would help to solve HC (and other problems).