Algorithms Homework #4

Due Jan 4, 2021 before class.

Collaboration policy: You can discuss the problem with other students, but you must obtain and write the final solution by yourself. Please specify all of your collaborators (name and student id) for each question. If you solve some problems by yourself, please also specify ”no collaborators”.

Problem 1 (15%)

n different houses are located along a road. The goal of this problem to build some fire stations to protect these houses. Each fire station must be built near one of the houses. Building a fire station near house i costs c_i and can cover houses $a_i, a_i+1, a_i+2, \ldots, b_i$. (You may assume that $a_i \leq i \leq b_i$.) Design an $O(n \log n)$-time algorithm to find the set of fire stations which covers all houses with minimum cost. Briefly justify the correctness and analyze the running time.

Problem 2 (15%)

Given n different currencies $c_1, c_2, \ldots, c_n$. 1 dollar in currency c_i can buy $w_{ij} > 0$ dollars in currency c_j (assuming no transaction fees). Given all exchange rates w_{ij}, we want to find out the best way to buy currency c_n starting with some money in currency c_1.

1. (5%) If the exchange rates are arbitrary and can be unrealistic, does the optimal exchange rate always exist? Why?
2. (10%) Given that the optimal exchange rate exists, design a polynomial time algorithm to find the optimal exchange rate. Briefly justify the correctness of your algorithm and analyze the running time.

Problem 3 (10%)

Given an undirected graph G in which every vertex has a positive weight. (Edges have weight 0.) Given two vertices s, t. Design an algorithm which finds the shortest path (= minimum total vertex weights) from s to t. Your algorithm must run in time $O(E + V \log V)$ in order to receive full credits. Briefly justify the correctness and analyze the running time.

Problem 4 (15%)

1. (8%) In Prim's algorithm, at each round, if there are multiple edges with equal minimum costs, the algorithm picks one of them arbitrarily. Consider the following modification: if adding all minimum-cost edges do not create cycles, the modified algorithm adds all of them to the output instead of just an arbitrary one. Is the modified algorithm still guaranteed to find the MST? Prove the correctness of the modified algorithm or find a counter example. (You do not have to worry about the running time and/or implementation and only need to focus on the correctness.)

2. (7%) Repeat the above problem for Kruskal's algorithm.

Problem 5 (15%)

Given a flow network with source s and sink t. We add an extra restriction that the maximum amount of flow that can go through vertex v_i is at most d_i (i.e. the total incoming flow to v_i is at most d_i). Given all edge capacities c_e and all vertex capacities d_i, design a polynomial time algorithm to find the maximum flow from s to t. Briefly justify the correctness and analyze the running time.

For problems 6 and 7, you must either

1. design an algorithm with running time polynomial in n, briefly justify the correctness,

   or

2. prove that the problem is NP-complete.

You may use the fact that SAT, 3-SAT, Vertex Cover, Set Cover, Independent Set, Hamiltonian Path and Hamiltonian Cycle problems are all NP-complete. Also, you do not need to prove that problems 5, 6 are in NP.

For problems 6 and 7, a conference program committee has n members and need to review m papers. The i-th member of the committee can only review papers in a subset S_i, for $i = 1, 2, \ldots, n$. Given the list of all members, all papers and all subsets S_i:

Problem 6 (15%)

Determine whether it is possible to select k members in the committee such that each paper has at least three selected members who can review it.

Problem 7 (15%)

Determine whether it is possible to assign each paper to exactly three committee members such that each committee member receives at least ten papers which they can review.