Algorithms   Homework #3

Due Dec 7, 2020 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”. Homeworks without collaborator specification will not be graded.

Problem 1 (15%)

Given an undirected graph $G = (V, E)$ and two nodes $u, v \in V$, find the number of shortest paths from u to v. (In this problem, there is no edge cost, so shortest paths are the paths with the smallest number of edges.) Notice that you do not need to specify all shortest paths, just counting the number is sufficient. Your algorithm should run in time $O(V + E)$ in order to receive full credit. Briefly justify the correctness and analyze the running time.

Problem 2 (15%)

Given an undirected graph $G = (V, E)$. Three different markers are initially placed at vertices s_1, s_2 and s_3 respectively. Each round, you may pick a pair of adjacent vertices u, v and either move all markers at vertex u to vertex v or move exactly one marker at vertex u to vertex v. Your goal is to move the three markers to vertices t_1, t_2 and t_3 respectively. Design a polynomial-time algorithm which finds the minimum number of rounds. Briefly justify the correctness and analyze the running time.

Problem 3 (20%)

n different courses are offered by NTUEE. Each course has 0 to 3 other courses as prerequisites. Each course can only be taken after all prerequisites are completed and takes one semester to finish. Assume that all courses are offered every semester and a person can take any number of courses in one semester. Design an algorithm to find the minimum number of semesters required to finish all of these n courses (given that you can complete all courses in some orderings). You algorithm must run in $O(n)$ time in order to receive full credit. Briefly justify the correctness and analyze the running time.

Problem 4 (15%)

Given a graph $G = (V, E)$ and a starting vertex $s \in V$. Design an $O(V + E)$ algorithm that marks all vertices reachable from s using a path (not necessarily a simple path) with the number of edges being multiples of 3. Your algorithm must be able to handle both directed and undirected graphs. Briefly justify the correctness and analyze the running time.

Problem 5 (20%)

The diameter of a graph is $\max_{u,v} d(u, v)$, where $d(u, v)$ is the number of edges on the shortest path from u to v. In other words, the diameter is the largest of all shortest-path distances in the graph. Given an (undirected) tree T with n nodes, design an $O(n)$ algorithm to compute the diamater of T. Formally prove the correctness of your algorithm and briefly analyze the running time.

Problem 6 (15%)

Given the graph (map) of a region consisting of n nodes (cities) and m edges (roads). The n cities are labeled with $v_1, v_2, \ldots, v_n$. Each road e has a (possibly different) driving time w_e. A person wants to drive from v_1 to v_n. He may take a one hour rest every time he passes through a city. He does not have to rest at every city he passes through but he can only drive at most k hours between any two rests. Given that all w_e and k are integers. Design an algorithm which finds the shortest trip (driving time and resting time combined). Your algorithm must run in time $O(k^2(m + n))$ in order to receive full credits. Briefly justify the correctness and analyze the running time.