COL 702

Homework II

Due on Sept. 5, 2019

Justify your answers with proper reasonings/proofs.

1. Let G be a directed graph. Let $C_1, C_2, \ldots, C_k$ denote the strongly connected components of G (each C_i is a subset of vertices). We now construct another directed graph H as follows: there are k vertices in H, denoted $v_1, \ldots, v_k$. There is a directed edge (v_i, v_j) in H if and only if there is an edge in G from a vertex in C_i to a vertex in C_j. Prove that H is a DAG. Give a linear time algorithm to construct H.

2. Call a directed graph G to be *weakly* connected if for every pair of vertices u, v , either there is a directed path from u to v or a directed path from v to u. Give a linear time algorithm to check if a directed graph is weakly connected.

3. Describe an efficient algorithm to find the second minimum shortest path between vertices u and v in a weighted graph without negative weights. The second minimum weight path must differ from the shortest path by at least one edge and may have the same weight as the shortest path.

4. You are given a directed graph G and two vertices s and t in G. Each edge of G is associated with a cost $c(e)$ that may be negative; however there is no negative cycle in G. Give an efficient algorithm to compute the total number of shortest s-t paths in G. Note that you are not supposed to output the set of such paths, but just a count of how many different shortest paths are there from s to t.

5. You are given a directed graph G where all edge lengths are positive except for one edge. Given a source vertex s, give $O(m \log n)$ time algorithm for finding a shortest path from a vertex s to a vertex t. Now assume there are a constant number of edges in G which have negative weights (rest have positive weights). Give an $O(m \log n)$ time algorithm to find a shortest path from s to t.

6. Let G be a directed graph and s be a vertex in it. Suppose you are given a tree rooted at s which contains all the vertices and such that all edges in it are directed away from s. Give an $O(m + n)$ time algorithm to check if this is a shortest path tree (you can assume that all edge lengths are non-negative).

7. You are given an undirected graph G. For an edge e let $G \setminus e$ denote the graph obtained by removing the edge e from G. You are now given two vertices s to t in G. You would like to compute the shortest s-t path length in $G \setminus e$ for **all** the edges e in G. Given an $O(m \log n)$ time algorithm to solve this problem.