School of Computer Science

Carleton University

Name: **A.M.** Problem Set: **5**

Course: **COMP 3804** Weeks: **5**-6

Problems

Consult [1, 2, 3].

Problem 1. *Given an undirected connected graph $G = (V, E)$, in adjacency list representation, can it be decided within $O(|V| + |E|)$ time whether there is a path between two specific vertices x and y consisting of at most 50 edges, where $x, y \in V$?*

Problem 2. *Can you devise a faster algorithm for computing single source shortest path distances from the source vertex s in an undirected graph $G = (V, E)$ when all the edge weights are 1? (Think of an algorithm that runs in $O(|V| + |E|)$ time on a graph $G = (V, E)$.)*

Problem 3. *Let $G = (V, E)$ be a weighted directed graph, where the weight of each edge is a positive integer and is bounded by a number X. Show how shortest paths from a given source vertex s to all vertices of G can be computed in $O(X|V| + |E|)$ time.*

Problem 4. *There is a road network between cities given to you as an undirected graph, and the vertices are the cities. There is an edge between two vertices, if and only if there is a direct road (not going through any other city) between the corresponding two cities. The weight of an edge is the distance between the two cities. There is a proposal to add one new road to this network, and there is a list E' of pairs of cities between which the new road can be built. Each such potential road has an associated length (the distance between the cities). As a politician, you must decide which new road should be built to lead to the maximum decrease in the distance between two specific (favourite) cities, say s and t. Give an efficient algorithm for determining which edge $e \in E'$ should be chosen to lead to the maximum decrease in shortest path distance between s and t.*

Problem 5. *Dijkstra's SSSP algorithm works for directed graphs where the weights of the edges are positive, and we need to compute the shortest paths from the source vertex to all other vertices in the graph. What happens when some of the edges have negative weights. Try to consider the cases where the algorithm will fail and where the algorithm will still work.*

Problem 6. *Let T and S be two minimum cost spanning tree of a graph $G = (V, E)$ on n vertices. Let $c_1 \leq c_2 \leq ... \leq c_{n-1}$ be the costs of the edges in T, and let $d_1 \leq d_2 \leq ... \leq d_{n-1}$ be the costs of the edges in S. Show that that $c_i = d_i$, for $1 \leq i \leq n - 1$.*

Problem 7. *How can we find a longest path (in terms of the number of edges) in a DAG $G = (V, E)$ in $O(|V| + |E|)$ time?*

References

[1] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms.* 3rd. ed., MIT Press, 2009.

[2] S. DasGupta, C. Papadimitriou, V. Vazirani. *Introduction to Algorithms.* McGraw Hill.

[3] A. Maheshwari. *Notes on Algorithm Design*, Chapter 1, `https://people.scs.carleton.ca/~maheshwa/Notes/DAA/notes.pdf`