CSOR S4231 – Summer 2018

Homework 3

Out: Saturday, June 9, 2018
Due: 10pm, Tuesday, June 19, 2018

Please keep your answers clear and concise. For all algorithms **you** *suggest, you must prove correctness and give the best upper bound that you can for the running time. You should always describe your algorithm clearly in English* **and** *give pseudocode.*

You should write up the solutions entirely on your own. Collaboration with your classmates is limited to discussion of ideas only. You should list your collaborators on your write-up. If you do not type your solutions, make sure that your hand-writing is legible, your scan is high-quality and your name is clearly written on your homework.

Notational convention: $|V| = n$, $|E| = m$.

1. (20 points) Given an undirected graph $G = (V, E)$ and two nodes $v, w \in V$, give an $O(n+m)$ algorithm that computes *the number* of shortest v-w paths in G.

2. (20 points) Problem 24-3, part a. only, from your textbook (p. 679).

3. (20 points) In cases where there are several shortest paths between two nodes (and edges have varying lengths), the most convenient among these paths is often the one with the *fewest* edges. We define

   $$best[u] = \text{minimum number of edges in a shortest path from } s \text{ to } u$$

   Give an efficient algorithm that, on input a weighted graph $G = (V, E, w)$ with positive edge weights, and an origin node $s \in V$, computes $best[u]$ for all $u \in V$.

4. (20 points) Consider a network of roads $G = (V, E, \ell)$ connecting a set of cities V, where each road in $e \in E$ has an associated length ℓ_e. There is a proposal to add one new road on this network, and there is a list E' of pairs of cities between which the new road can be built. Each such potential road $e' \in E'$ has an associated length $\ell_{e'}$.

   As a designer for the public works department you are asked to determine the road $e' \in E'$ whose addition to the existing network G will result in the maximum decrease in the driving distance between two *fixed* cities s and t in the network. Given an efficient algorithm that solves this problem.

5. (20 points) In the shortest-paths algorithm we are concerned with the *total length* of the path between a source (origin) node s and every other node. Suppose instead that we are concerned with the length of the *longest edge* in a path between the source node and every node. That is, the *bottleneck* of a path is defined to be the length of the longest edge in the path.

   Design an efficient algorithm to solve the single-source smallest bottleneck problem, i.e., find the paths from a source to every other node such that each path has the smallest possible bottleneck. (You may assume that the graph is undirected.)

OPTIONAL exercises (do NOT return, they will not be graded)

1. Problem 22-2, parts a, b, c, d, e and f, in your textbook (p. 622).

2. Problem 22-3 in your textbook (p. 623).

3. Show how to compute single-origin shortest paths in a DAG in time $O(n + m)$ (see Section 24.3, if needed).

4. Design an efficient algorithm to compute the *longest* path in a directed acyclic graph.