CS 4349.501 Homework 8

Due Wednesday November 1st, in class

October 25, 2017

Please answer the following questions. Each student must write their solutions in their own words and submit their solutions on paper at the beginning of class. ***Include your name and/or Net ID at the top of each page.*** And please staple your papers together if you can.

1. Let $G = (V, E)$ be a directed graph with edge weights $w : E \to \mathbb{R}$ (which may be positive, negative, or zero), and let s be an arbitrary vertex of G.

   (a) Suppose every vertex v stores a number $dist(v)$. Describe and analyze an algorithm that returns 'yes' if $dist(v)$ is the shortest-path distance from s to v for *every* vertex v. Otherwise, the algorithm should return 'no'. Your algorithm should be asymptotically faster than one that computes shortest path distances from scratch.

   (b) Suppose instead that every vertex $v \neq s$ stores a pointer $pred(v)$ to another vertex in G. Describe and analyze an algorithm that returns 'yes' if these predecessor pointers define a single-source shortest path tree rooted at s. Otherwise, your algorithm should return 'no'. Your algorithm should be asymptotically faster than one that computes shortest path distances from scratch.

   The running time for both parts should be in terms of V and E, the number of vertices and edges in the input graph.

2. Let $G = (V, E)$ be a directed graph with edge weights $w : E \to \mathbb{R}$ (which may be positive, negative, or zero), and let s be an arbitrary vertex of G.

   (a) Suppose edge $u \to v$ is tense after V phases of Shimbel's single-source shortest path tree algorithm. What is the length of the shortest *walk* from s to v? Don't forget to explain your answer.

   (b) Describe and analyze a modification of Shimbel's algorithm that computes the correct shortest path distance from s to every other vertex of G, even if the graph contains negative cycles. Specifically, if any walk from s to v contains a negative cycle, your algorithm should end with $dist(v) = -\infty$; otherwise, $dist(v)$ should contain the length of the shortest path from s to v. The modified algorithm should still run in $O(VE)$ time.