

## Homework 4

## Graph Algorithms

Due: Friday, May 12, 11:45pm

Please answer the following questions, each of which is worth 10 points.

1. (CS 163 students only) Using the graph at the following website, label each vertex with its closeness centrality.

<https://i.stack.imgur.com/kJF5K.png>

2. (CS 163 students only) Using the graph at the following website, label each vertex with its betweenness centrality.

<https://i.stack.imgur.com/kJF5K.png>

3. A certain Professor Amongus claims that there is an easier way to reweight edges than the method used in Johnson's algorithm. His method begins by defining  $\hat{w}$  to be the minimum (possibly negative) edge in the input graph,  $G = (V, E)$ . That is,  $\hat{w} = \min_{(u,v) \in E} \{w(u, v)\}$ . Professor Amongus claims that you can skip the initial Bellman-Ford step and simply replace each edge weight,  $w(u, v)$ , with a new weight  $w'(u, v) = w(u, v) - \hat{w}$ , and then just do the Dijkstra steps. Prove or disprove whether Professor Amongus is correct.
4. Give an example of a graph  $G$  with at least 10 vertices such that the greedy 2-approximation algorithm for VERTEX-COVER given in class is guaranteed to produce a suboptimal vertex cover.
5. Give a complete, weighted graph,  $G$ , such that its edge weights satisfy the triangle inequality but the MST-based approximation algorithm for TSP does not find an optimal solution.
6. (CS 265 students only) In the *bottleneck traveling salesperson problem*, we are given an undirected graph  $G$  with weights on its edges and asked to find a tour that visits the vertices of  $G$  exactly once and returns to the start so as to minimize the cost of the maximum-weight edge in the tour. Assuming that the weights in  $G$  satisfy the triangle inequality, design a polynomial-time 3-approximation algorithm for bottleneck TSP.
7. (CS 265 students only) In the *Euclidean traveling salesperson problem*, cities are points in the plane and the distance between two cities is the Euclidean distance between the points for these cities, that is, the length of the straight line joining these points. Show that an optimal solution to the Euclidean TSP is a simple polygon, that is, a connected sequence of line segments such that no two ever cross.