CS 4349.006 Homework 8

Due Thursday, November 4 on eLearning

Please solve the following **2** problems.

1. Suppose you are given both an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$ and a minimum spanning tree T of G.

   (a) Describe and analyze an algorithm to update the minimum spanning tree when the weight of a single edge e is increased.

   Advice: Do you need to do anything if $e \notin T$? If $e \in T$, argue that $T - e$ is a subgraph of the new minimum spanning tree. How do you find its last edge?

   (b) Describe and analyze an algorithm to update the minimum spanning tree when the weight of a single edge e is decreased.

   Advice: Do you need to do anything if $e \in T$? If $e \notin T$, argue that the new minimum spanning tree is a subgraph of $T + e$. Graph $T + e$ contains a single cycle; which edge should you remove? (See Homework 7, Problem 2.)

   For both parts, the input to your algorithm includes the edge e and its new weight; your algorithm should modify T so that it is still a minimum spanning tree. Your algorithms for both parts should run in $O(E)$ time.

2. Although we often speak of "the" shortest path between two vertices, a single graph could contain several minimum-length paths with the same endpoints. Even for weighted graphs, it is often desirable to choose a minimum-weight path with the fewest edges; call this path a ***best path*** from s to t. Suppose we are given a directed graph $G = (V, E)$ with positive, zero, or negative integer edge weights $w : E \to \mathbb{Z}$ and a source vertex s in G. Describe and analyze an algorithm to compute *best* paths from s to every other vertex.

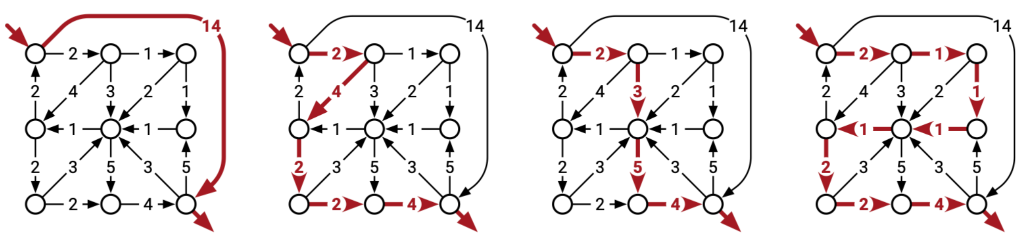

Figure 1. Four (of many) equal-weight shortest paths. The first path is the "best" shortest path.

Advice: **Do not** *modify a shortest paths algorithm from class. Instead, modify the edge weights and call an unmodified shortest paths algorithm.*