Version: 1.0

- Suppose we are given both an undirected graph G with weighted edges and a minimum spanning tree T of G.
  - **1.A.** Describe an efficient algorithm to update the minimum spanning tree when the weight of one edge  $e \in T$  is decreased.
  - **1.B.** Describe an efficient algorithm to update the minimum spanning tree when the weight of one edge  $e \notin T$  is increased.
  - **1.C.** Describe an efficient algorithm to update the minimum spanning tree when the weight of one edge  $e \in T$  is increased.
  - **1.D.** Describe an efficient algorithm to update the minimum spanning tree when the weight of one edge  $e \notin T$  is decreased.

In all cases, the input to your algorithm is the edge e and its new weight; your algorithms should modify T so that it is still a minimum spanning tree. Of course, we could just recompute the minimum spanning tree from scratch in  $O(|E| + |V| \log |V|)$  time, but you can do better.

Let G = (V, E) be an undirected graph where each edge has a weight from the set  $\{1, 10, 25\}$ . Describe a linear-time algorithm to find an MST of G.