Intro to Algorithms, COMP-160, Homework #11
Benjamin Tanen, 04/23/2016

1. **You are given a graph G and its MST. Do not assume that all edge weights are distinct. Suppose the weight of some edge e in G is decreased. Show how to update the MST. Carefully analyze the time-complexity of your algorithm, and prove correctness.**

   For this problem, we have two cases to consider. The first case is if the edge e is already in our MST M. Since e is already in M, we know there is no edge x from G where $w(x) < w(e)$ if x isn't already in M (since we were guaranteed M at the start). Therefore, even if we reduce the weight of e, we know that M is still our MST. Thus in order to maintain M, we do nothing which takes $\Theta(1)$ time.

   The other case is if the edge e is not already in M. In order to find our MST in this case, we simply add e to M. Since we had $V - 1$ edges in M before (by definition of an MST), with the inclusion of e, we will now have V edges.

   The inclusion of e will also create a loop in M that contains the end-points of our edge e. We know this because in our original MST, there existed a path from e_1 to e_2 (the end-points of e). By including e, we formed a path from e_2 back to e_1 (simply traverse e) so we can see that a loop formed (note: this is the only loop added to M). In order to maintain our MST we need to remove one of the edges from this loop, specifically the edge with the largest weight.

   In order to do this, we can do a DFS by starting at e_1. Since we know there are V edges (including e), this search takes $O(V + V) = O(V)$. While we are searching, we can keep track of the largest weight w_{large} for any edge that we travel along. If in our search we hit a leaf, we can simply disregard the largest weight we've been tracking. If we hit e_1, we know that we found the loop inside of M and we are currently storing w_{large} for the edges of this loop.

   We can now backtrack along this loop path (from our DFS recursion) and when we traverse an edge that has the same weight as w_{large}, we simply delete this edge (since it has the largest weight). This backtrack takes $O(V)$ time. Since we previously had V edges before, deleting this largest edge makes it so that we now have $V - 1$ edges, a necessity of having an MST. Since the deleted edge was part of the loop, we know there exists another path from one end-point of this edge to another, thus maintain our MST.

   Thus, in total, the second case requires $O(V + V) + O(V) = O(V)$ time.