We first do this under the assumption that all edge costs are distinct. In this case, we can solve (a) as follows. Let $e = (v, w)$ be the new edge being added. We represent T using an adjacency list, and we find the v-w path P in T in time linear in the number of nodes and edges of T, which is $O(|V|)$. If every node on this path in T has cost less than c, then the Cycle Property implies that the new edge $e = (v, w)$ is not in the minimum spanning tree, since it is the most expensive edge on the cycle C formed from P and e, so the minimum spanning tree has not changed. On the other hand, if some edge on this path has cost greater than c, then the Cycle Property implies that the most expensive such edge f cannot be in the minimum spanning tree, and so T is no longer the minimum spanning tree.

For (b), we replace the heaviest edge on the v-w path P in T with the edge $e = (v, w)$, obtaining a new spanning tree T'. We claim that T' is a minimum spanning tree. To prove this, we consider any edge e' not in T', and show that we can apply the Cycle Property to conclude that e' is not in any minimum spanning tree. So let $e' = (v', w')$. Adding e' to T' gives us a cycle C' consisting of the v'-w' path P' in T', plus e'. If we can show e' is the most expensive edge on C', we are done.

To do this, we consider one further cycle: the cycle K formed by adding e' to T. By the Cycle Property, e' is the most expensive edge on K. So now there are three cycles to think about: C, C', and K. Edge f is the most expensive edge on C, and edge e' is the most expensive edge on K. Now, if the new edge e does not belong to C', then $C' = K$, and so e' is the most expensive edge on C'. Otherwise, the cycle K includes f (since C' needed to use e instead), and C' uses a portion of C (including e) and a portion of K. In this case, e' is more expensive than f (since f lies on K), and hence it is more expensive than everything on C (since f is the most expensive edge on C). It is also more expensive than everything else on K, and so it is the most expensive edge on C', as desired.

Now, if the edge costs are not all distinct, we apply the approach in the chapter: we first perturb all edge costs by extremely small amounts so they become distinct. Moreover, we do this so we add a very small quantity ϵ to the new edge e, and we perturb the costs of all other edges f by even much smaller, distinct, quantities δ_f. For a tree T, let $c(T)$ denote its real (original) cost, and let $c'(T)$ denote its perturbed cost.

Now we use the above solution with distinct edge costs. Our perturbation has the following two properties.

(i) First, for trees T_1 and T_2, if $c'(T_2) < c'(T_1)$, then $c(T_2) \leq c(T_1)$.

(ii) Second, if $c(T_1) = c(T_2)$, and T_2 contains e but T_1 doesn't, then $c(T_2) > c(T_1)$.

It follows from these two properties that our conclusion in (a) is correct: since $c'(T') < c'(T)$, and T' contains e but T doesn't, property (i) implies $c(T') \leq c(T)$, and then property (ii) implies $c(T') < c(T)$. Now, in (b), we compute a minimum spanning tree with respect to the perturbed costs which, by property (i), is also one of (possibly several) minimum spanning trees with respect to the real costs.

[1] ex833.93.54