Intro exercises for MST algorithms

1) Show that every graph $G = (V, E)$ that's a tree (an undirected graph without cycles) has $|E| = |V| - 1$. Hint: Induction.

Base case(s): For |V| = n = 1, 2, 3 nodes there is only a single tree graph (up to isomorphisms; each is a chain) containing 0, 1, 2 edges.

Induction: Let T(k) be |E| for a graph with k nodes. Assume T(k) = k – 1 for k <= n. We want to show T(n+1) = n. Any tree on n+1 nodes can be constructed from taking two smaller trees whose total nodes add up to n+1 and connecting them with a single edge. Let k1, k2 be the size of these trees; k1 + k2 = n+1. Thus, T(n+1) = T(k1) + T(k2) + 1 = k1 – 1 + k2 – 1 + 1 = n + 1 - 1 = n.

Another nice direct proof is to root the tree at an arbitrary vertex. Then every vertex except the root has a unique parent connected by a single edge. Thus |E| = |V| - 1.

2) Assuming distinct costs for each edge, explain why the costliest edge in any cycle cannot belong to any MST of the graph.

Let C be a cycle in a graph G with $e \in C$ the edge with largest weight in the cycle. Then *e* cannot belong to any MST of G.

Proof by contradiction: Assume *e* actually is part of some MST. Removing *e* from the MST disconnects the graph into two components. However, there exists an edge *e'* in the cycle that isn't used and will connect the graph. Reconnect the graph using that edge *e*'. Now it is a spanning tree with smaller weight since w(*e*') < w(*e*). Since this contradicts that we started with an MST in the first place, *e* cannot be part of any MST.