Section 1.3.3

1 Let G be a connected graph. Assign weight 1 to all edges of G, and run Kruskal's algorithm. The algorithm produces a minimal spanning tree, thus G contains at least one minimum spanning tree.

2 $\Rightarrow$ Suppose G is a tree. Remember that $|E(G)| = n-1$ where $n = |V(G)|$. G is connected by definition, and contains all its vertices so G is a spanning tree of G which contains all its edges. If G contained more than one spanning tree, (other than G itself), than these two spanning trees must differ on at least one edge, meaning $|E(G)| > n-1$, a contradiction.

$\Leftarrow$ Suppose a graph G is connected and contains exactly one spanning tree S. We show that $G = S$. S is a subgraph of G, thus $V(S) \subseteq V(G)$ and $E(S) \subseteq E(G)$. If there exists $v \in V(G), v \notin V(S)$, then S is not a spanning tree of G, thus $V(G) \subseteq V(S)$. Suppose $\exists e \in E(G), e \notin E(S)$. We give weight of 0 to e, and weight 1 to all other $e' \in E(G)$, and run Kruskal's algorithm, producing a minimum spanning tree T. $T \neq S$, since $e \in T$ since the algorithm will always choose the lowest weight edge, but $e \notin S$ by assumption, leading to a contradiction that G has only one minimum spanning tree. Thus $E(G) \subseteq E(S)$, and $G = S$.

3 Let T be a spanning tree of G. Suppose $\overline{T}$ does not contain any edges in C. Then T contains C, thus T is not a tree.

4 We prove the contrapositive of the two statements.

$\Rightarrow$ Suppose $e \in E(G)$ and there exists a spanning tree T such that $e \notin E(T)$. Then T spans $G-e$, meaning $\forall u, v \in V(G), \exists uv$ path in T, so $G-e$ is connected.

$\Leftarrow$ Suppose e is not a bridge. Then $G-e$ is connected, and thus has a spanning tree T. This is a spanning tree of G that doesn't contain e.