2.1.37 Let $T, T'$ be two spanning trees of a connected graph $G$. For $e \in E(T) - E(T')$, prove that there is an edge $e' \in E(T') - E(T)$ such that $T' + e - e'$ and $T - e + e'$ are both spanning trees of $G$.

Assume $T, T'$ are two distinct spanning trees of connected graph $G$ with $n$ vertices. Let $e \in E(T) - E(T')$, and suppose $e = uw$. Then $T - e$ has exactly two components. Let $U$ and $W$ be the vertex sets of these components with $u \in U$ and $w \in W$. Since $T'$ is a tree, it has a unique $u, w$-path. There must be an edge of the path that has an endpoint in $U$ and an endpoint in $W$. Choose such an edge and call it $e'$.

- $T - e + e'$ is a spanning tree: it is connected (every vertex in $U$ is connected to an endpoint of $e'$ and every vertex in $W$ is connected to the other endpoint of $e'$) and has $n - 1$ edges. Note that $e' \notin E(T)$ since its endpoints are in two different components of $T - e$.

- $T' + e - e'$ is a spanning tree: since $e'$ is on the unique cycle formed by adding $e$ to $T'$, we know that $T' + e - e'$ is acyclic and has $n - 1$ edges.

(Side-note: We could have labeled the set $W$ as $U$ since $W = V(G) - U = \overline{U}$.)
Let $G$ be a connected graph with $n$ vertices. Define a new graph $G'$ having one vertex for each spanning tree of $G$, with vertices adjacent in $G'$ if and only if the corresponding trees have exactly $n - 2$ common edges. Prove that $G'$ is connected. For $n \geq 4$, determine the maximum possible value of $\text{diam}(G')$ as a function of $n$. An example appears below.

* Note that this problem is slightly modified from how it is stated in your textbook.

The diameter of a graph $G$, denoted $\text{diam}(G)$, is the maximum length of a shortest $(u, v)$-path over all pairs of vertices $u, v$. In other words, $\text{diam}(G) = \max_{u, v \in V(G)} d(u, v)$, where $d(u, v)$ is the length of a shortest $(u, v)$-path.

We prove that $G'$ is connected with a proof by induction on $|E(G)|$. Let the number of vertices, $n$, be fixed. (Since $G$ is a connected graph on $n$ vertices, we know that $|E(G)| \geq n - 1$, so the base case is $|E(G)| = n - 1$.)

Base case: Suppose $G$ is a connected graph with $n$ vertices and $n - 1$ edges. Then $G$ is a tree, so it has exactly one spanning tree (itself) and $G' = K_1$, which is clearly connected.

Induction step: Assume that for any graph $G$ on $n$ vertices and $m \geq n - 1$ edges, we have that $G'$ is connected. Now consider a graph $G$ on $n$ vertices with $m + 1$ edges.

Since $G$ has at least $n$ edges ($m + 1 \geq (n - 1) + 1 = n$), it follows that $G$ has a cycle $C$. Let $e \in E(C)$. The graph $G - e$ remains connected since $e$ is not a cut edge, and by the induction hypothesis, $(G - e)'$ is connected. Every spanning tree of $G - e$ is itself a spanning tree of $G$, so $(G - e)'$ is an induced subgraph of $G'$.

Since $(G - e)'$ is connected, to show that $G'$ is connected, it suffices to show that every spanning tree $T$ of $G$ that does contain $e$ is adjacent in $G'$ to a spanning tree $T'$ of $G$ that does not contain $e$. By Proposition 2.1.6 (or Problem 2.1.37), there exists an edge $e' \in E(T') - E(T)$ such that $T - e + e'$ is a spanning tree of $G$. Since $T$ and $T' = T - e + e'$ are two spanning trees of $G$ with $n - 2$ edges in common, they are adjacent in $G'$, and so $T'$ is a spanning tree that does not contain $e$ that is adjacent to $T$. We conclude that $G'$ is connected.

Claim: The diameter of $G'$ is at most $n - 1$, with equality holding when $G$ has two spanning trees that share no edges.
It suffices to show that the distance between two spanning trees $T$ and $T'$ in $G'$ is $|E(T) - E(T')|$, i.e., $\text{dist}_{G'}(T, T') = |E(T) - E(T')|$. Each edge on a path from $T$ to $T'$ in $G'$ discards at most one edge of $T$, so the distance is at least $|E(T) - E(T')|$. Furthermore, since for each edge $e \in E(T) - E(T')$, there exists $e' \in E(T') - E(T)$ such that $T - e + e' \in V(G')$, the path built in the previous argument (for connectedness of $G'$) has precisely this length.

In addition, since trees with $n$ vertices have $n - 1$ edges, we know that $
 |E(T) - E(T')| \leq n - 1$, so

$$\text{diam}(G') = \max_{T, T'} \text{dist}_{G'}(T, T') \leq n - 1.$$  

When $G$ has two edge-disjoint spanning trees, the diameter of $G'$ is exactly $n - 1$. For example, consider $K_n$ for $n \geq 4$ and assume its vertices are labeled $v_1, v_2, \ldots, v_n$. Then two edge-disjoint spanning trees $T, T'$ are described by the following edge sets:

$$E(T) = \{v_1v_2, v_2v_3, \ldots, v_{n-1}v_n\}$$

$$E(T') = \begin{cases} 
\{v_1v_3, v_3v_5, \ldots, v_{n-2}v_n\} \cup \{v_2v_4, \ldots, v_{n-3}v_{n-1}\} \cup \{v_2v_n\} & \text{if } n \text{ odd} \\
\{v_1v_3, v_3v_5, \ldots, v_{n-3}v_{n-1}\} \cup \{v_2v_4, \ldots, v_{n-2}v_n\} \cup \{v_1v_n\} & \text{if } n \text{ even} 
\end{cases}$$

[2.2.6] Let $G$ be the 3-regular graph with $4m$ vertices formed from $m$ pairwise disjoint kites by adding $m$ edges to link them in a ring, as shown below for $m = 6$. Prove that $\tau(G) = 2m8^m$.

Let us call the edges joining kites the “link edges” of $G$. Observe that deleting two link edges disconnects the graph, so every spanning tree of $G$ omits at most one link edge.

We consider two cases.

- **Suppose a spanning tree uses $m - 1$ link edges.** Then any spanning tree for each kite may be used to form a spanning tree of $G$. By Example 2.2.6, the kite has 8 possible spanning trees. Thus, there are $m$ choices for which link edge to omit and 8 choices of which spanning tree for each of the $m$ kites. In total, there are $m8^m$ spanning trees of this type.
Suppose a spanning tree uses all \( m \) edges. Then to avoid having a cycle, there must be one kite in \( G \) that is spanned by a disconnected spanning subgraph rather than by a spanning tree. (However, there cannot be more then one such kite, since this would lead to a disconnected spanning subgraph of \( G \).) The kite has 8 disconnected spanning subgraphs that will prohibit a cycle from occurring in the spanning tree of \( G \) (\( u, v \) are vertices of degree 3 within a given kite):

So there are \( m \) choices for which kite has a disconnected spanning subgraph, 8 choices for the spanning subgraph on the selected kite, and 8 possibilities for the spanning trees on the \( m - 1 \) other kites. Therefore, there are \( m \cdot 8 \cdot 8^{m-1} = m8^m \) spanning trees of this type.

Given the two cases above, the total number of spanning trees of \( G \) is

\[
\tau(G) = m8^m + m8^m = 2m8^m
\]

2.2.7 Use Cayley’s Formula to prove that the graph obtained from \( K_n \) by deleting an edge has \((n-2)n^{n-3}\) spanning trees.

By Cayley’s Formula, we have \( \tau(K_n) = n^{n-2} \). We count the number of pairs \((e, T)\) such that \( T \) is a spanning tree of \( K_n \) and \( e \in E(T) \). Since there are \( n - 1 \) edges in any spanning tree, we have that there are \((n - 1)n^{n-2}\) such pairs. Group these pairs according to the \( \binom{n}{2} \) edges of \( K_n \), i.e., if the edges of \( K_n \) are labeled \( e_1, e_2, \ldots, e_m \) where \( m = \binom{n}{2} \), then define groups

\[
S_i = \{(e_i, T) : T \text{ is a spanning tree}\}
\]

for \( 1 \leq i \leq \binom{n}{2} \). Because of the symmetry of \( K_n \), every edge of \( K_n \) appears in the same number of spanning trees, which implies that \( |S_i| = |S_j| \) for all \( i, j \). So

\[
\text{number of spanning trees of } K_n \text{ containing any given edge } = \frac{\sum_i |S_i|}{\binom{n}{2}} = \frac{(n-1)n^{n-2}}{n(n-1)/2} = 2n^{n-3}.
\]

To count the spanning trees of \( K_n - e \) for a particular edge \( e \in E(K_n) \), we subtract from the total number of trees in \( K_n \) the number that contain \( e \). This results in

\[
n^{n-2} - 2n^{n-3} = (n-2)n^{n-3}.
\]
2.2.15 Let $G_n$ be the graph with $2n$ vertices and $3n - 2$ edges pictured below, for $n \geq 1$. Prove for $n > 2$ that $\tau(G_n) = 4\tau(G_{n-1}) - \tau(G_{n-2})$.

Proof 1:
Using the deletion/contraction formula, we have the following computation. (Note that for each graph $G$ shown in the computation below, we actually mean $\tau(G)$.)

The edges colored green indicate edges which must be included in all spanning trees of the graph and so can be effectively disregarded when counting the number of spanning trees of the graph. On the other hand, edges colored red indicate edges which will not be included in any spanning trees of the graph and so also can be disregarded when counting spanning trees.

In the last, far right step, we note that the graph under consideration (after the red loop is deleted) is $G_{n-1} \cdot e$. By the deletion/contraction formula, we know that

$$\tau(G_{n-1}) = \frac{\tau(G_{n-1} - e) + \tau(G_{n-1} \cdot e)}{\tau(G_{n-2})} \implies \tau(G_{n-1} \cdot e) = \tau(G_{n-1}) - \tau(G_{n-2}).$$

By the deletion/contraction formula for $\tau(G)$, we have

$$\tau(G_n) = 3\tau(G_{n-1}) + (\tau(G_{n-1}) - \tau(G_{n-2})) = 4\tau(G_{n-1}) - \tau(G_{n-2}).$$
Proof 2:

Note that any spanning tree $T$ of $G_n$ must include 2 or 3 of the edges in the set $\{e_1, e_2, e_3\}$. We consider two cases.

**Case 1:** Consider all spanning trees that contain exactly 2 edges from among $e_1, e_2, e_3$. Notice that
\[
\text{number of spanning trees of } G_n \text{ that use } e_1, e_2 = \text{number of spanning trees of } G_n \text{ that use } e_1, e_3 = \text{number of spanning trees of } G_n \text{ that use } e_2, e_3 = \tau(G_{n-1}).
\]

Thus the number of spanning trees of $G_n$ that contain exactly 2 edges from among $e_1, e_2, e_3$ is $3\tau(G_{n-1})$.

**Case 2:** Now consider all spanning trees of $G_n$ that contain all 3 edges $e_1, e_2, e_3$. Let $T$ be such a spanning tree. Since $T$ is acyclic, we know $e \not\in T$ and further that $T - e_1 - e_2 - e_3 + e$ is a spanning tree of $G_{n-1}$. Moreover, any spanning tree of $G_{n-1}$ that contains $e$ can be transformed into a spanning tree of $G_n$ that contains $e_1, e_2, e_3$ by adding in these edges and removing $e$. Since we have a one-to-one correspondence between spanning trees of $G_n$ that contain $e_1, e_2, e_3$ and spanning trees of $G_{n-1}$ that do contain $e$, we know these two sets must be of equal size.

So we have
\[
\text{number of spanning trees of } G_n \text{ that contain } e_1, e_2, e_3 = \text{number of spanning trees of } G_{n-1} \text{ that do contain } e = \text{number of spanning trees of } G_{n-1} - \text{number of spanning trees of } G_{n-1} \text{ that do not contain } e.
\]

But the quantity on the right-hand side of the last equality above is precisely $\tau(G_{n-1}) - \tau(G_{n-2})$.

Since we have partitioned the set of all spanning trees of $G_n$ into two parts and counted each of those parts in the cases above, we conclude that $\tau(G_n)$ is the sum of the two results found above, i.e.,
\[
\tau(G_n) = 4\tau(G_{n-1}) - \tau(G_{n-2}).
\]
2.3.7 Let \( G \) be a weighted connected graph with distinct edge weights. Without using Kruskal’s algorithm, prove that \( G \) has only one minimum weight spanning tree. (Hint: Use Exercise 2.1.37.)

Suppose, for sake of contradiction, that \( G \) has two distinct minimum-weight spanning trees \( T \) and \( T' \). Then let \( e \) be the edge lowest weight in the symmetric difference, \( E(T) \triangle E(T') \). Without loss of generality, assume \( e \in E(T) \). Since \( e \in E(T) - E(T') \), by Problem 2.1.37, there exists an edge \( e' \in E(T') - E(T) \) such that \( T' + e - e' \) is a spanning tree. By the choice of \( e \) and the fact that all edge weights of \( G \) are distinct, we know that \( w(e) < w(e') \). It follows that

\[
    w(T' + e - e') = w(T') + w(e) - w(e') < w(T'),
\]

contradicting the assumption that \( T' \) is a minimum spanning tree. Hence, \( G \) can have only one minimum spanning tree.

2.3.10 Prim’s Algorithm grows a spanning tree from a given vertex of a connected weighted graph \( G \), iteratively adding the cheapest edge from a vertex already reached to a vertex not yet reached, finishing when all the vertices of \( G \) have been reached. (Ties are broken arbitrarily.) Prove that Prim’s Algorithm produces a minimum-weight spanning tree of \( G \).

Let \( T \) be the tree produced by Prim’s algorithm with \( v \) as the initial vertex in the algorithm. By way of contradiction, assume \( T \) is not a minimum weight spanning tree. Let \( T^* \) be an optimal tree that agrees with \( T \) for the most steps (given that we started from vertex \( v \)). Let \( e \) be the first edge chosen for \( T \) that does not appear in \( T^* \), and let \( U \) be the set of vertices in the subtree of \( T \) that has been grown prior to the addition of \( e \). Adding \( e \) to \( T^* \) creates a cycle \( C \); since \( e \) links \( U \) to \( \overline{U} \), \( C \) must contain another edge \( e' \) from \( U \) to \( \overline{U} \).

Since \( T^* + e - e' \) is another spanning tree, the optimality of \( T^* \) implies \( w(e') \leq w(e) \). Since \( e' \) is incident to \( U \), \( e' \) is available for consideration when \( e \) is chosen by the algorithm; since the algorithm chose \( e \), we have \( w(e) \leq w(e') \). Hence \( w(e) = w(e') \), and \( T + e - e' \) is a spanning tree with the same weight as \( T^* \). This means it is an optimal spanning tree that agrees with \( T \) longer than \( T^* \), which contradicts the choice of \( T^* \).

Therefore, it must be that, in fact, \( T \), the tree produced by Prim’s algorithm, is a minimum spanning tree.

2.3.14 Let \( C \) be a cycle in a connected weighted graph. Let \( e \) be an edge of maximum weight on \( C \). Prove that there is a minimum weight spanning tree not containing \( e \). Use this to prove that iteratively deleting a heaviest non-cut-edge until the remaining graph is acyclic produces a minimum weight spanning tree.
Let $T$ be a minimum spanning tree of a connected weighted graph $G$, and assume $e$ is an edge of maximum weight in a cycle $C$ of $G$. If $e \in E(T)$, then $T - e$ has two components with vertex sets $U$ and $\overline{U}$. The subgraph $C - e$ of $G$ is a path with endpoints in $U$ and $\overline{U}$. Thus, it must contain an edge $e'$ that has an endpoint in $U$ and an endpoint in $\overline{U}$. Since $e$ was a max. weight edge in $C$, we know that $w(e') \leq w(e)$, which implies that $T - e + e'$ is a spanning tree of $G$ whose weight is at most that of $T$. So $T - e + e'$ is a minimum spanning tree of $G$ that does not use $e$.

A heaviest non-cut edge is a heaviest edge on a cycle (since all non-cut edges occur in cycles). We have shown that some minimum spanning tree avoids such an edge, so deleting it from $G$ does not change minimum weight of a spanning tree. This remains true as we continue to delete heaviest edges of cycles. When no cycles remain, we have a connected, acyclic spanning subgraph. It is the only remaining spanning tree and has the minimum weight among spanning trees of the original graph.

■