The argument in the text does show that for any 38. Let \( P(k) \) be: every tree with \( k \) edges has at least two leaves. We will prove this by strong induction for all \( k \geq 1 \). This, together with the fact that a tree with 0 edges clearly has 1 leaf, proves that every tree has a leaf.

Basis \( P(1) \). Such a tree consists of a (nonloop) edge, so both ends are leaves.

Now assume all \( P(j) \) for \( 1 \leq j < k \). Consider any tree \( T \) with \( k \) edges and delete any one edge \( e \). This disconnects \( T \) into two components, each of which is a tree (since they still have no cycles). By assumption, each of these trees, \( T_1 \) and \( T_2 \), has at least 2 leaves. Put \( e \) back in. The ends of \( e \) will not be leaves, even if they were in \( T_1 \) and \( T_2 \), but this leaves at least \( 4 - 2 = 2 \) leaves in \( T \). QED

Note: the typical build-up error is to start with a tree with \( k - 1 \) edges and add a new edge dangling off of some vertex. But to assume that every tree with \( k \) edges can be constructed that way is to assume at the start the very thing you are supposed to prove — because the other end of that new edge is automatically a leaf.

39. The argument in the text does show that for any \( n \in N^+ \) there is some alkane of the form \( C_nH_{2n+2} \), because it shows how to construct at least one such alkane for each \( n \), by induction. But is every alkane (that is, every acyclic, simple graph with all vertices of degree 4 or 1) obtainable by this construction? If not, we haven’t shown that all alkanes have the formula \( C_nH_{2n+2} \). If yes, an additional argument is needed to show it. In other words, this argument has a build-up error.

Fortunately, the error is fixable — every alkane can be obtained this way. But the proof is a bit subtle. Let \( P(n) \) be the claim that every alkane with \( n \) carbons can be built from the unique \( C_1H_4 \) by repeated replacing an \( H \) with a \( C_2H_3 \) group. \( P(1) \) is clearly true. To show \( P(n) \implies P(n+1) \), we must take any alkane with \( n+1 \) carbons and show how it can be obtained by replacing a hydrogen in some \( n \)-carbon alkane by a \( C_1H_3 \) group, for by \( P(n) \) that \( n \)-carbon alkane can be built up this way. The key observation is that the subgraph of carbons within any alkane is itself a tree (why?). So consider our arbitrary \( (n+1) \)-carbon alkane and pick a leaf in this tree. That carbon must be part of a \( C_1H_3 \) group (why?). Replace it by an \( H \). Now we have an \( n \)-carbon alkane. Reversing the process shows that the arbitrary \( (n+1) \)-carbon alkane we started with can be obtained as claimed.

40. The flaw in the proof results because the removal of one vertex may disconnect the graph so the “remaining graph” may not have a spanning tree. Fortunately, it is always possible to remove a particular vertex which does not disconnect a disconnected graph. Why? If the graph is acyclic (a tree), then there is a vertex of degree 1, a leaf (see [31, Section 3.6]). The removal of this vertex will not disconnect the graph (why?). If the graph is not acyclic, it contains a spanning tree, and removing a vertex that is a leaf in the spanning tree will again not disconnect the rest, because they are connected by the rest of the tree. But wait! This reasoning is circular — we have used the very result the induction was meant to prove. So, if we really want an independent inductive proof that every graph has a spanning tree, we do not see how to fix it by showing that some vertex can be removed without disconnecting the graph.

Fortunately, there is another fix. Don’t worry whether \( G - v \) is connected or not. By strong induction, every one of its components has a spanning tree, and every one of its components has an edge from it to \( v \). The union of all these little spanning trees and all these connecting edges is a spanning tree of \( G \) (why?)