The following is an attempt at a detailed induction proof about the number of internal and external nodes of a binary tree. The theorem itself is true, but this particular approach to proving it doesn't work.

Theorem: For a binary tree with n nodes ($n > 0$), the number of internal nodes, plus one, is greater than or equal to the number of leaves.

Restatement: For any tree T, let $I(T)$ be the number of internal nodes, and $E(T)$ be the number of external nodes (i.e., leaves). We'll show that for a binary tree with n nodes, I(T) + 1 >= E(T).

Proof (by induction): Let P(k) be the statement that for any tree U with k nodes, $I(U) + 1 >= E(U)$.

P(1) is evidently true: the only one-node tree has one internal node and no external nodes.

Now we'll suppose that P(k) is true for some unspecified number $k > 1$, and will show that this implies that $P(k + 1)$ is also true.

Let T be a tree with $k + 1$ nodes; we need to show that I(T) + 1 >= E(T).

Since T has at least two nodes, it certainly has a leaf, v, that is not the root; delete v and the edge to its parent (which exists because the leaf is not the root) and call the result U. Then U has k nodes, and we can apply $P(k)$ to conclude that

(*) $I(U) + 1 >= E(U)$.

There are two possibilities: the parent of v has two children in T, or it has only one child (v).

In the first case, every internal node of U is an internal node of T and vice versa, so $I(U) = I(T)$. So we have, by substitution in (*),

(**) $I(T) + 1 >= E(U)$

And every leaf of U is a leaf of T, but there's one additional leaf (v) in T, so $E(T) - 1 = E(U)$. Substituting this in (**) gives us

(***) $I(T) + 1 >= E(T) – 1$

so

        $I(T) + 2 >= E(T)$

Unfortunately, at this point we're stuck. This isn't what we wanted to prove, and there's no easy way to see how to proceed forward. Sometimes an approach to induction just doesn't pan out…