1.(a) Assume $N(h)$ be number of nodes in a complete binary tree.
Base Case:
When the height of the tree is 0, then $N(h) = 2^{0+1} - 1 = 1$, which is true. Since the complete binary tree has 1 node if the height of the tree is 0.

Inductive hypothesis:
Assume the total number of a complete binary tree is $N(h) = 2^{h+1} - 1$ for $h \geq 0$.

Induction Steps:
We know that a complete binary tree consists of 2 complete binary subtrees and the root. Each subtrees has height of $h - 1$. Thus,

$$\begin{aligned} N(h) &= 2 \times N(h-1) + 1 \\ &= 2 \times (2^{h-1+1} - 1) + 1 \\ &= 2 \times (2^h - 1) + 1 \\ &= 2 \times 2^h - 2 + 1 \\ &= 2^{h+1} - 1 \end{aligned}$$

Then, we want to prove it is true for the height of $h + 1$. We know that for each height of h, we have 2^h leaf nodes. For the height of $h + 1$ the number of leaf node increases to 2×2^h (or 2^{h+1}) since each leaf node at the height h has 2 children. Then,

$$\begin{aligned} N(h+1) &= N(h) + 2^{h+1} \\ &= 2^{h+1} - 1 + 2^{h+1} \\ &= 2 \times (2^{h+1}) - 1 \\ &= 2^{h+2} - 1 \end{aligned}$$

Hence, a complete binary tree with height h contains $2^{h+1} - 1$ total nodes.

1.(b) To prove a complete binary tree with n nodes has $\frac{n-1}{2}$ internal nodes is to prove that the number of total nodes n in a complete binary tree is $2 \times internalnodes(I) + 1$.

Base Case:
When $I = 0, n = 2 \times 0 + 1 = 1$, which is true since the tree only has the root. When $I = 1, n = 3$, which is obviously true since the tree has a root and its children.

Inductive hypothesis:
Assume a complete binary tree with I internal nodes has $2I + 1$ nodes.

Induction Steps: We want to prove that a complete binary tree with $I + 1$ internal nodes has $2(I + 1) + 1$ total nodes.

Let T be a complete binary tree with $I+1$ internal nodes. T has at least one internal nodes and at least two leaves. Select a leaf l at maximal height of the tree and remove leaf l and its sibling. That builds a new complete binary tree T'. The parent of l is an internal nodes in T but a leaf in T'. T' has I internal nodes and contains $2I+1$ total nodes by assumption. Thus, T with $I+1$ internal nodes has $(2I+1)+2$ total nodes since it has l and its sibling. Hence, the number of total nodes n in a complete binary tree is $2 \times internalnodes(I) + 1$. Which means a complete binary tree with n nodes has $\frac{n-1}{2}$ internal nodes.

2. Assume y is not the ancestor of x, then we let z denote as the first common ancestor of x and y. Due to the property of binary search tree, $x < z < y$, which means y cannot be the successor of x. Thus, y must be an ancestor of x,and it is the lowest ancestor of x. If there exists z denote as the lowest ancestor of x, then z must be the left subtree of y , and $z < y$. That means z will be the successor of x. In order to let y be the successor of x, y must be the lowest ancestor of x.
   Assume the left child of y is not an ancestor of x. Then the right child of y must be an ancestor of x, implying $x > y$. Hence, the left child of y is an ancestor of x.