3.

Let LE represent “Left subtree of node x is empty”

Let XP represent “x has a predecessor y in the tree”

Let YR represent “y has a right child”

Need to prove: $(LE \wedge XP) \rightarrow YR$

There are 3 cases for this problem.

1) X is the root node.
2) X is left child of a node.
3) X is right child of a node.

1) Given x is the root node, all keys in the tree are distinct, and left subtree of x is empty. In this case, x is the smallest key in the tree, since no key is the same with x. Hence, x does not have any predecessor in this case.

$$(LE \wedge XP) \rightarrow YR \equiv (LE \wedge F) \rightarrow YR \equiv F \rightarrow YR \equiv T$$

2) Given x is left child of a node, and left subtree of x is empty, then parent(x) $\geq$ x and right(x) $\geq$ x. But since all keys in the tree are distinct, parent(x) > x and right(x) > x. This means that x is still the smallest key in the tree. So, x does not have any predecessor.

$$(LE \wedge XP) \rightarrow YR \equiv (LE \wedge F) \rightarrow YR \equiv F \rightarrow YR \equiv T$$

3) Given x is the right child of a node, and left subtree of x is empty. We have parent(x) $\leq$ x. But since all keys in tree are distinct, we parent(x) < x. In this case, parent(x) is the largest key smaller than x, and hence p(x) = y. Since x is the right child of parent(x), x is the right child of y, and y indeed has a right child.

$$(LE \wedge XP) \rightarrow YR \equiv T \wedge T \rightarrow T \equiv T$$

Hence, all 3 cases suffices to prove that In a binary search tree in which the keys are all distinct, if the left subtree of node x is empty and x has a predecessor y in the tree, then y has a right child.