CMSC 15300 Homework problem (Due 5/28/2003)

Consider the following definition of binary trees labeled with integers (in SML syntax):

```sml
datatype tree = Leaf | Nd of (tree * int * tree);
```

Values of this type include:

- Leaf
- Nd(Leaf, 1, Leaf)
- Nd(Nd(Leaf, 1, Leaf), 5, Nd(Leaf, 1, Leaf))

We define the `values` of a tree inductively as follows:

- `values(Leaf) = {}`
- `values(Nd(t1, x, t2)) = \{x\} \cup values(t1) \cup values(t2)`

**Binary search trees** are trees in which an inorder traversal produces an increasing sequence of node labels. We can formalize this property with the following definition:

```sml
BST(t) \equiv (t = Nd(t1, x, t2)) \Rightarrow [\forall y(y \in values(t1) \Rightarrow (y < x)) \land \forall y(y \in values(t2) \Rightarrow (x < y))]
```

We can use a binary search tree to represent sets of integers. The following SML function tests to see if its first argument is a member of the set represented by its second argument:

```sml
fun member (x, t) = (case t of Leaf => false
                        | Nd(t1, y, t2) => if (x < y) then member(x, t1)
                                       else if (x > y) then member(x, t2)
                                       else true
                        (* end case *))
```

Prove, by induction, the following correctness statement:

```sml
\forall t : tree [BST(t) \Rightarrow (member(x, t) \Leftrightarrow x \in values(t))]```