Is the binary tree a search tree?

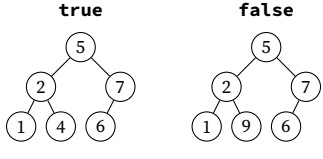
Suppose you are given a binary tree B of unique integers. If v is a node in the binary tree, then

- v.`element` is the integer stored in the node
- v.`left` is the left child of v
- v.`right` is the right child of v

We want to check if the binary tree is also a binary *search tree*. Below you will find the specification for the algorithm, and two examples of trees that respectively should return **`true`** and **`false`**.

Input: The root node v of a binary tree B
Output: Returns **`true`** if the binary tree is a binary search tree, **`false`** otherwise

```
1 Procedure CheckBST(v)
  |   // ···
```



There are several good solutions to this problem. The following can be helpful when thinking of a solution:

- You can assume that you have procedures `FindMin` and `FindMax` which respectively finds the smallest and largest number in the binary tree. Since the tree is not guaranteed to be a binary search tree (or balanced) then these procedures will have linear time.

- It may be a good idea to split the algorithm by creating a help procedure.

(a) Write down the property a binary tree must have to be called a binary *search tree*.

(b) Complete the procedure above. A lower runtime complexity gives a higher score.

(c) Give the runtime complexity of your algorithm with respect to the number of nodes in the binary tree.