98. Validate Binary Search Tree

Given a binary tree, determine if it is a valid binary search tree (BST).

Assume a BST is defined as follows:

- The left subtree of a node contains only nodes with keys **less than** the node's key.
- The right subtree of a node contains only nodes with keys **greater than** the node's key.
- Both the left and right subtrees must also be binary search trees.

confused what `"{1,#,2,3}"` means? > read more on how binary tree is serialized on OJ.

All values on the left subtree must be less than the root and all right subtree must be greater than the root. SO we just check the boundaries for each node.
Complexity: Time O(N) Space O(1)

```python
19  class Solution(object):
20      # @param root, a tree node
21      # @return a boolean
22
23      def isValidBST(self, root):
24          positivInf = float('inf')
25          negativeInf = float('-inf')
26          return self.rec(root, negativeInf, positivInf)
27
28      def rec(self, node, min, max):
29          if node == None:
30              return True
31
32          if node.val <= min or node.val >= max :
33              return False
34
35          return self.rec(node.left, min, node.val) and self.rec(node.right, node.val, max)
```