Difficulty: **Category:** **Successful Submissions:** 37,237+

Validate BST

Write a function that takes in a potentially invalid Binary Search Tree (BST) and returns a boolean representing whether the BST is valid.

Each `BST` node has an integer `value`, a `left` child node, and a `right` child node. A node is said to be a valid `BST` node if and only if it satisfies the BST property: its `value` is strictly greater than the values of every node to its left; its `value` is less than or equal to the values of every node to its right; and its children nodes are either valid `BST` nodes themselves or `None` / `null`.

A BST is valid if and only if all of its nodes are valid `BST` nodes.

Sample Input

```
tree =   10
       /     \
      5       15
    /   \   /   \
   2     5 13   22
 /           \
1             14
```

Sample Output

```
true
```

Hints

Hint 1

Every node in the BST has a maximum possible value and a minimum possible value. In other words, the value of any given node in the BST must be strictly smaller than some value (the value of its closest right parent) and must be greater than or equal to some other value (the value of its closest left parent).

Hint 2

Validate the BST by recursively calling the validateBst function on every node, passing in the