

[Description](#)
[Solution](#)
[Discuss \(999+\)](#)
[Submissions](#)

173. Binary Search Tree Iterator

Medium 4835 352 Add to List Share

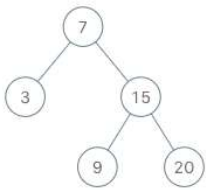
Implement the `BSTIterator` class that represents an iterator over the **in-order traversal** of a binary search tree (BST):

- `BSTIterator(TreeNode root)` Initializes an object of the `BSTIterator` class. The `root` of the BST is given as part of the constructor. The pointer should be initialized to a non-existent number smaller than any element in the BST.
- `boolean hasNext()` Returns `true` if there exists a number in the traversal to the right of the pointer, otherwise returns `false`.
- `int next()` Moves the pointer to the right, then returns the number at the pointer.

Notice that by initializing the pointer to a non-existent smallest number, the first call to `next()` will return the smallest element in the BST.

You may assume that `next()` calls will always be valid. That is, there will be at least a next number in the in-order traversal when `next()` is called.

Example 1:



Input

```
["BSTIterator", "next", "next", "hasNext", "next", "hasNext", "next",  
"hasNext", "next", "hasNext"]
```

```
[[[7, 3, 15, null, null, 9, 20]], [], [], [], [], [], [], [], [], []]
```

Output

```
[null, 3, 7, true, 9, true, 15, true, 20, false]
```

Explanation

```
BSTIterator bSTIterator = new BSTIterator([7, 3, 15, null, null, 9,
20]);
```

```
bSTIterator.next(); // return 3
bSTIterator.next(); // return 7
bSTIterator.hasNext(); // return True
bSTIterator.next(); // return 9
bSTIterator.hasNext(); // return True
bSTIterator.next(); // return 15
bSTIterator.hasNext(); // return True
bSTIterator.next(); // return 20
bSTIterator.hasNext(); // return False
```

Constraints:

- The number of nodes in the tree is in the range $[1, 10^5]$.
- $0 \leq \text{Node.val} \leq 10^6$
- At most 10^5 calls will be made to `hasNext`, and `next`.

Follow up:

- Could you implement `next()` and `hasNext()` to run in average $O(1)$ time and use $O(h)$ memory, where h is the height of the tree?

Accepted 498,483 Submissions 770,040

Seen this question in a real interview before?

Yes No

Companies

Related Topics

Similar Questions

i Java

Autocomplete

```

1  /**
2   * Definition for a binary tree node.
3   * public class TreeNode {
4   *     int val;
5   *     TreeNode left;
6   *     TreeNode right;
7   *     TreeNode() {}
8   *     TreeNode(int val) { this.val = val; }
9   *     TreeNode(int val, TreeNode left, TreeNode right) {
10    *         this.val = val;
11    *         this.left = left;
12    *         this.right = right;
13    *     }
14    * }
15    */
16    class BSTIterator {
17
18    public BSTIterator(TreeNode root) {
19
20    }
21
22    public int next() {
23
24    }
25
26    public boolean hasNext() {
27
28    }
29    }
30
31    /**
32     * Your BSTIterator object will be instantiated and called as such:
33     * BSTIterator obj = new BSTIterator(root);
34     * int param_1 = obj.next();
35     * boolean param_2 = obj.hasNext();
36     */

```