[**Binary Search Tree Iterator**](https://leetcode.com/problems/binary-search-tree-iterator/)

**/\*\***

**\* Definition for a binary tree node.**

**\* public class TreeNode {**

**\* int val;**

**\* TreeNode left;**

**\* TreeNode right;**

**\* TreeNode(int x) { val = x; }**

**\* }**

**\*/**

**class BSTIterator {**

**int index;**

**List<Integer> list;**

**public BSTIterator(TreeNode root) {**

**this.index = -1;**

**this.list = new ArrayList<>();**

**this.inorder(root);**

**}**

**public void inorder(TreeNode root) {**

**if(root == null)**

**return;**

**inorder(root.left);**

**list.add(root.val);**

**inorder(root.right);**

**}**

**/\*\* @return the next smallest number \*/**

**public int next() {**

**return list.get(++index);**

**}**

**/\*\* @return whether we have a next smallest number \*/**

**public boolean hasNext() {**

**return index + 1 < list.size();**

**}**

**}**

**/\*\***

**\* Your BSTIterator object will be instantiated and called as such:**

**\* BSTIterator obj = new BSTIterator(root);**

**\* int param\_1 = obj.next();**

**\* boolean param\_2 = obj.hasNext();**

**\*/**

Time Complexity : O(n) , n is no of nodes in BST

Space Complexity : O(n), n is no of nodes in BST