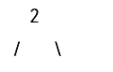
Binary Search Tree

1. Given the root of a binary tree, check whether it is a BST or not.
   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 equal or greater than the node's key.
- Both the left and right subtrees must also be binary search trees.

Input:

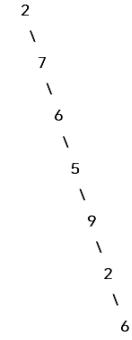


Output: 1

Explanation:

The left subtree of root node contains node with key lesser than the root nodes key and the right subtree of root node contains node with key greater than the root nodes key. Hence, the tree is a BST.

Input:



Output: 0

Explanation:

Since the node with value 7 has right subtree nodes with keys less than 7, this is not a BST.

2. Given a sorted array. Write a function that creates a Balanced Binary Search Tree using

Input: nums = {1,2,3,4,5,6,7}

Ouput: {4,2,1,3,6,5,7}

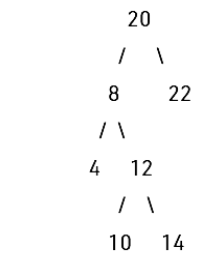
Explanation:

The preorder traversal of the following BST formed is {4,2,1,3,6,5,7} :

array elements. Height balanced BST means a binary tree in which the depth of the left subtree and the right subtree of every node never differ by more than 1.

3. Given a BST, and a reference to a Node x in the BST. Find the Inorder Successor of the given node in the BST.

Input:



K(data of x) = 8

Output: 10

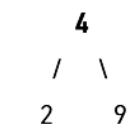
Explanation:

Inorder traversal: 4 8 10 12 14 20 22

Hence, successor of 8 is 10.

4. Given a Binary search tree, your task is to complete the function which will return the Kth largest element without doing any modification in the Binary Search Tree.

Input:



k = 2

Output: 4