

DAILY ONLINE ACTIVITIES SUMMARY

Date:	21-06-2020	Name:	Deepika K V
Sem & Sec	8 th sem 'A' sec	USN:	4AL16CS030
Online Test Summary			
Subject	--		
Max. Marks	-	Score	-
Certification Course Summary			
Course	Learn Projects in Javascript and JQuery		
Certificate Provider	eduonix	Duration	9 hrs
Coding Challenges			
Problem Statement: Write a python program to check the given binary tree is a valid binary search tree(BST) or not.			
Status: SUBMITTED			
Uploaded the report in Github		YES	
If yes Repository name		Codes	
Uploaded the report in slack		YES	

Online test details:

Certification Course Details:

The screenshot displays the Eduonix website interface. The top navigation bar includes the Eduonix logo, an 'Explore' button, a search bar with the placeholder 'What you want to learn today?', and links for 'LIFETIME MEMBERSHIP' and 'OFFER ZONE'. The user is logged in as 'Hi Deepika'. The main content area features a large banner for 'Projects Using JavaScript & jQuery' with the subtitle 'Project 1 Intro - Simple JavaScript Quiz'. Below the banner, the text 'Project 1 Intro' is displayed, followed by 'From the course: Learn Projects In JavaScript And JQuery' and a 'Generate Certificate' button. On the right side, a sidebar shows the 'Contents' tab selected, listing 'All Lectures (50)'. The list includes '1: Introduction' (1/1 Lectures Completed) and '2: Simple JavaScript Quiz' (5/5 Lectures Completed). Under '2: Simple JavaScript Quiz', there are six items: '1 Course Intro', '2 Project 1 Intro' (highlighted), '3 JavaScript Fundamentals', '4 Quiz File Structure and HTML', '5 Quiz CSS', and '6 Quiz JavaScript Logic'. Each item has a green checkmark and a folder icon.

Coding Challenge:

```
INT_MAX =
4294967296

INT_MIN = -4294967296

class Node:

    # Constructor to create a new node
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None

    # Returns true if the given tree is a binary search tree
    # (efficient version)
    def isBST(node):
        return (isBSTUtil(node, INT_MIN, INT_MAX))
```

```

# Return true if the given tree is a BST and its values
# >= min and <= max
def isBSTUtil(node, mini, maxi):

    # An empty tree is BST
    if node is None:
        return True

    # False if this node violates min/max constraint
    if node.data < mini or node.data > maxi:
        return False

    # Otherwise check the subtrees recursively
    # tightening the min or max constraint
    return (isBSTUtil(node.left, mini, node.data -1) and
            isBSTUtil(node.right, node.data+1, maxi))

# Driver program to test above function
root = Node(4)
root.left = Node(2)
root.right = Node(5)
root.left.left = Node(1)
root.left.right = Node(3)

if (isBST(root)):
    print ("Is BST")
else:
    print ("Not a BST")

```