QUIZ 2 ON TREE (set 16)
Time: 15 mins
Mark: 10

1. Consider the following numbers:

[20, 9, 16, 31, 35, 25, 23, 17, 4, 12, 27]

a) Draw **Binary Search Tree** by inserting the above numbers as keys of the nodes from left to right. **[2 marks]**

b) Write the **post-orde**r traversal sequence of the tree and compute the **depth** of **node 16**. **[2 mark]**

c) Identify whether the tree is full, complete, perfect and balanced. **[1 mark]**

d) Perform the following operations step by step on the Binary Search Tree you created in part 'a'. **[2 marks]**

i. Delete node with the key of value **31** with the help of its **predecessor**.

ii. Delete node with the key of value **20** with the help of its **successor**.

2. Given the array representation of a binary tree: [**null value means the node is empty**]
[Null, 71, 50, 90, 20, Null, Null, 98, Null, 40, Null, Null, Null, Null, 94, Null]

a) Draw the **binary tree** and compute the **height** of the tree.**(Root is 71)** **[2 marks]**

a) Convert the tree to a **complete binary tree**. (hint: remove all Null from the array and start creating the nodes by following the rules of root node being at index 1) **[1 mark]**

QUIZ 3 ON RECURSION (set 16)
Time: 10 mins
Mark: 10

1. Construct a recursive method which takes an array and index as an input and prints all the **negative numbers which are divisible by 3** in **reversed** order.

```
def reverse(arr, idx):
        #write code
```

Sample input:
arr=[4, 17, 1, -3, -15, 6, -9]

Tester Code:
reverse(arr, idx=0)
Output: -9, -15, -3 -> since these are negative numbers divisible by 3