18-March-2020

Link for practice:

https://thecodingsimplified.com/binary-tree/

Topics Covered:

Content:
18-March-2020

1. Create Binary tree - Done
2. https://www.geeksforgeeks.org/binary-tree-set-2-properties/
3. Sum of all Nodes - Done
4. Get Total Number of Nodes - Done
5. Get Number of Leaf Nodes - Done
6. Height of a Binary Tree - Done
7. Print Elements at a given level - Done

H.W.:

1. Practice class Questions
2. Level Order
3. Level Order Line By Line
4. Level Order using queue

Code

```
package mbatch;
class Node
{
	int data;
	Node left;
	Node right;
	Node(int data){
		this.data=data;
		left=null;
		right=null;
	}
}
class BinaryTree{
	Node root;
	BinaryTree(){
		root=null;
	}
	BinaryTree(int data){
		this.root=new Node(data);
	}
	int TreeSum(Node root)//to calculate the sum of all nodes
in a tree
	{
		if(root==null) return 0;
		return
root.data+TreeSum(root.left)+TreeSum(root.right);
	}
	int countNodes(Node root)//to calculate the number of nodes
in a tree
	{
		if(root==null) return 0;
		return 1+countNodes(root.left)+countNodes(root.right);
	}
	int leafNodes(Node root)//to calculate the number of Leaf
Nodes in a tree
	{
		if(root==null) return 0;
		if(root.left==null && root.right==null) return 1;
		return leafNodes(root.left)+leafNodes(root.right);
	}
	int height(Node root)
	{
		if(root==null) return -1;
```

```
            return 1+Math.max(height(root.left),
height(root.right));
      }
      void printAtLevel(Node root,int level)
      {
            if(root==null) return;
            if(level==1)
            {
                  System.out.print(root.data+" ");
                  return;
            }
            printAtLevel(root.left,level-1);
            printAtLevel(root.right,level-1);
      }
}
public class btree {
      public static void main(String[] args) {
            // TODO Auto-generated method stub
            BinaryTree bt=new BinaryTree(2);//BT with root node 2
            bt.root.left=new Node(3);//linking explicitly
            bt.root.right=new Node(5);
            bt.root.left.right=new Node(9);
            bt.root.right.left=new Node(7);//Required Tree Created
            System.out.println("Sum of all Nodes:
"+bt.TreeSum(bt.root));
            System.out.println("Total Nodes:
"+bt.countNodes(bt.root));
            System.out.println("Leaf Nodes:
"+bt.leafNodes(bt.root));
            System.out.println("Height: "+bt.height(bt.root));
            System.out.print("Nodes at level 1: ");
            bt.printAtLevel(bt.root,1);
            System.out.println();
            System.out.print("Nodes at level 2: ");
            bt.printAtLevel(bt.root,2);
            System.out.println();
            System.out.print("Nodes at level 3: ");
            bt.printAtLevel(bt.root,3);
            System.out.println();
            System.out.print("Nodes at level 4: ");
            bt.printAtLevel(bt.root,4);
            System.out.println();
      }

}
```