Assignment #05

Due: At start of class on 18 April 2023

You are provided with the code for the tree ADT, contains add node and display functionalities along with an iterator. You may skip the understanding of iterator, if finding difficulties.

1. Understand the code of Tree and its driver's main logic. There is a bug in the **addnode** logic, at least identify it, better to rectify it.

2. Create a member function to return the **size** of the tree. Size being the count of nodes in the tree.

```
def size(self):
```

3. Create a member function **leaves** to return the count of leaf nodes in the tree.

```
def leaves(self):
```

4. Create a member function **update** to update the data of the node having **o** with a new value **n**.

```
def update(self, o, n):
```

5. Create a member function **level_wise** to display the tree though iterative logic using Queue instead of Stack.

```
def level_wise(self):
```

6. If depth of a node is defined as the number of edges from the root to the node. Create a member function **depth** to return depth of the node with data **n** stored in it.

```
def depth(self, n):
```

7. Create a member function **cut_paste** to move a subtree with node having data **o** from its parent and add as child node of **n**.

```
def cut_paste(self, o, n):
```

8. Create a member function **remove** to delete a subtree with node having data **v** in it.

```
def remove(self, v):
```

==========