create a binary tree class that accepts

-
    a. insertion
    b. traversal

```
class Mytree:
    class Nodetree:
        def __init__(self,data):
            self.data=data
            self.left=None
            self.right=None
    def __init__(self):
        self.root=None
    def insert(self,data):
        new_node=Mytree.Nodetree(data)
        if self.root==None:
            self.root = new_node
            return
        current = self.root
        while True:
            if data< current.data:
                if current.left is None:
                    current.left = new_node
                    break
                else:
                    current =current.left
            else:
                if current.right is None:
                    current.right = new_node
                    break
                else:
                    current =current.right
    def search(self,val):
        current = self.root
        while current:
            if current.data == val:
                return True
                break
            elif val<current.data:
                current =current.left
            else:
                current=current.right
        return False
    def in_order_traversal(self,node):
        if node:
            self.in_order_traversal(node.left)
            print(node.data, end=' ')
```

```
            self.in_order_traversal(node.right)

    def pre_order_traversal(self,node):
        if node:
            print(node.data, end=' ')
            self.pre_order_traversal(node.left)
            self.pre_order_traversal(node.right)
    def post_order_traversal(self,node):
        if node:
            self.post_order_traversal(node.left)
            self.post_order_traversal(node.right)
            print(node.data, end=' ')



m=Mytree()

m.insert(11)
m.insert(9)
m.insert(12)
m.insert(17)
m.insert(10)

m.search(12)

True

m.search(67)

False

m.in_order_traversal(m.root)

9 10 11 12 17

m.pre_order_traversal(m.root)

11 9 10 12 17

m.post_order_traversal(m.root)

10 9 17 12 11

def fact(val)
```