```
#implement python program to demonstrate binary search tree it's
operation

class Node:
    def __init__ (self,key):
        self.left=None
        self.right=None
        self.val=key

def insert (root,key):
    if root is None:
        return Node(key)
    else:
        if root.val==key:
            return root
        elif root.val<key:
            root.right=insert(root.right,key)
        else:
            root.left=insert(root.left,key)
    return root
def Inorder(root):
    if root:
        Inorder(root.left)
        print(root.val,end=" ")
        Inorder(root.right)

def Preorder(root):
    if root:
        print(root.val,end=" ")
        Preorder(root.left)
        Preorder(root.right)

def Postorder (root):
    if root:
        Postorder(root.left)
        Postorder(root.right)
        print(root.val,end=" ")

#Driver code
if __name__=="__main__":
    r=Node(25)
    r=insert(r,20)
    r=insert(r,36)
    r=insert(r,10)
    r=insert(r,22)
    r=insert(r,30)
    r=insert(r,40)
    r=insert(r,5)
    r=insert(r,12)
    r=insert(r,28)
    r=insert(r,38)
    r=insert(r,48)

#function call
```

```
print("Inorder traversal of binary search tree is :")
Inorder(r)
print("\n")
print("\n Preorder traversal of binary search tree is :")
Preorder(r)
print("\n")
print("\n Postorder traversal of binary search tree is:")
Postorder(r)
print("\n")
```

```
********************************************output*****************************************
Inorder traversal of binary search tree is :
5 10 12 20 22 25 28 30 36 38 40 48


 Preorder traversal of binary search tree is :
25 20 10 5 12 22 36 30 28 40 38 48


 Postorder traversal of binary search tree is:
5 12 10 22 20 28 30 38 48 40 36 25
```