

Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING



(Autonomous College Affiliated to the University of Mumbai) NAAC Accredited with "A" Grade (CGPA: 3.18)

Academic Year: 2022-2023

Name:	Prerna Sunil Jadhav
Sap Id:	60004220127
Class:	T. Y. B. Tech (Computer Engineering)
Course:	Advance Algorithm Laboratory
Course Code:	DJ19CEL602
Experiment No.:	04-A

AIM: Implement Red-black Tree Operations.

04-A) INSERTION

CODE:

```
# RB tree insertion
class Node:
 def __init__(self, val, color):
    self.val = val
    self.color = color
    self.left = None
    self.right = None
    self.parent = None
class RedBlackTree:
    def __init__(self):
        self.root = None
    def insert(self, val):
        new_node = Node(val, "RED")
        if not self.root:
            self.root = new_node
            new_node.color = "BLACK"
            return
        curr = self.root
        parent = None
        while curr:
            parent = curr
            if val < curr.val:</pre>
                curr = curr.left
            else:
                curr = curr.right
        new_node.parent = parent
        if val < parent.val:</pre>
            parent.left = new_node
        else:
```

Shri Vile Parle Kelavani Mandal's



DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

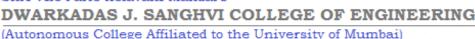


(Autonomous College Affiliated to the University of Mumbai) NAAC Accredited with "A" Grade (CGPA: 3.18)

Academic Year: 2022-2023

```
parent.right = new_node
       self._fix_violations(new_node)
   def fix violations(self, node):
       while node.parent and node.parent.color == "RED":
           if node.parent == node.parent.parent.left:
               uncle = node.parent.parent.right
               if uncle and uncle.color == "RED":
                   node.parent.color, uncle.color, node.parent.parent.color =
"BLACK", "BLACK", "RED"
                   node = node.parent.parent
               else:
                   if node == node.parent.right:
                       node = node.parent
                       self._left_rotate(node)
                   node.parent.color, node.parent.parent.color = "BLACK",
"RED"
                   self._right_rotate(node.parent.parent)
           else:
               uncle = node.parent.parent.left
               if uncle and uncle.color == "RED":
                   node.parent.color, uncle.color, node.parent.parent.color =
"BLACK", "BLACK", "RED"
                   node = node.parent.parent
               else:
                   if node == node.parent.left:
                       node = node.parent
                       self._right_rotate(node)
                   node.parent.color, node.parent.parent.color = "BLACK",
"RED"
                   self._left_rotate(node.parent.parent)
       self.root.color = "BLACK"
   def left rotate(self, node):
       right_child = node.right
       node.right = right_child.left
       if right child.left:
           right_child.left.parent = node
       right_child.parent = node.parent
       if not node.parent:
```

Shri Vile Parle Kelavani Mandal's





NAAC Accredited with "A" Grade (CGPA: 3.18)

Academic Year: 2022-2023

```
self.root = right_child
        elif node == node.parent.left:
            node.parent.left = right_child
        else:
            node.parent.right = right_child
        right child.left = node
        node.parent = right_child
    def _right_rotate(self, node):
        left child = node.left
        node.left = left_child.right
        if left child.right:
            left_child.right.parent = node
        left child.parent = node.parent
        if not node.parent:
            self.root = left child
        elif node == node.parent.right:
            node.parent.right = left_child
        else:
            node.parent.left = left child
        left_child.right = node
        node.parent = left_child
    def inorder_traversal(self, node):
        if node:
            self.inorder traversal(node.left)
            print(f"{node.val} ({node.color})", end=" ")
            self.inorder_traversal(node.right)
# Example usage
tree = RedBlackTree()
for val in [8,18,5,15,17,25,40,80]:
  tree.insert(val)
print("Inorder traversal of Red Black Tree:");
tree.inorder traversal(tree.root)
```

OUTPUT:

```
PS C:\Users\Jadhav\Documents\BTech\Docs\6th Sem\AA\Code> \& C:\msys64/mingw64/bin/python.exe "c:\Users\Jadhav\Documents) Additional Code of the Code 
Inorder traversal of Red Black Tree:
PS C:\Users\Jadhav\Documents\BTech\Docs\6th Sem\AA\Code>
```