# Insertion in Red-Black Tree (Writeup)

Amleez Mishra

IBM 18 CS013

```
Node insert (int v, Node root)
{      Node newNode;

   • if ( root == null )
        {  calor = 1; // black
           return newNode;
        }
        2.
        Node r = root;
        while ( r != null )
        {
           if ( r.val < v )
               @ root = insert (v, root.right )
           else
               root = insert (v, root.left)
        }

        r.parent = newNode.parent;
        if ( r.val > v )
               r.right = new Node;
        else
               r.left = new Node;

   calor = 0 // red.
```

Amber Mishra

1BM18CS013

fix After Insertion (root &p , new Node)

```
Node    fix After Insertion ( Node root, Node new Node)
{
    Node p = null  // parent  of new Node

    Node gp = null // grand-parent  of new Node

    if ( gp . left == p )
    {
        if ( gp . right . color == 0)
        {
            gp . left . color = 1;
            gp . left . color = 1;
            p new Node . grandparent = gp;
        }
    }
    else if ( p . right == new Node)
    {
        new Node . parent = p
        leftRotate ( new Node)
        p . color = 1
        gp . color = 0
        rightRotate ( gp )
    }
}
```

else
{

```
newNode.grandParent = gp;
if ( .p.left == newNode)
  { newNode.parent = p
    rightRotate (newNode);
  }
  p.color = 1;
  gp.color = 0;
  do leftRotate (gp)
  }
root.color = 1;
}
```