Red - Black Trees

Keshna Prasad V
1BM17CS047
SA

Assuming RB Tree class with a node

fixviolation (Node k)

```
{
    Node v
    while ( k -> parent -> color == 1)
    { if (k -> parent == k -> parent -> parent -> right)
      { v = k -> parent -> parent -> left
        if (v -> color == 1)
        { v -> color = 0
          k -> parent -> color = 0
          k -> parent -> parent -> color = 1
          k = k -> parent -> parent
        }
        else
        { if (k == k -> parent -> left)
          { k = k -> parent
            rightRotate (k)
          }
          k -> parent -> color = 0
          k -> parent -> parent -> color = 1
          leftRotate (k -> parent -> parent)
        }
      }
      else
      { v = k -> parent -> parent -> right
        if (v -> color == 1)
        { v -> color = 0
          k -> parent -> color = 0
          k -> parent -> parent -> col = 1
```

```
            k = k->parent->parent;
        }
        else
        {   if (k == k->parent->left)
            {   k = k->parent;
                leftRotate(k);
            }
            k->parent->color = 0;
            k->parent->parent->color = 1;
            rightRotate(k->parent->parent);
        }
        if (k == root)
            break;
    }
    root->color = 0;
}
insert(int key)
{   Node node = new Node();
    node->parent = null;
    node->data = key;
    node->left = TNULL;
    node->right = TNULL;
    node->color = 1;

    Node y = null;
    Node x = this->root;
```

```
while (x != NULL)
{   y = x
    if (node->data < x->data)
        x = x->left
    else
        x = x->right
}
node->parent = y
if (y == null)
    root = node
else if (node->data < y->data)
    y->left = node
else
    y->right = node
if (node->parent == null)
{   node->color = 0
    return
}
if (node->parent->parent == null)
    return
fixviolation (node)
}
```