Prathik Karanth
1BM18CS139

```
void insertfix (Node k)
   Node u;
   while (k-> ~~parent~~ parent -> color == 1)
  { if (k-> parent == k-> parent -> parent -> right)
   { u = k-> parent -> parent -> left;
     if (u-> color == 1) {
        u-> color = 0;
        k-> parent -> color = 0;
        k-> parent -> parent -> color = 1;
        k = k-> parent -> parent;
     } else {
       if (k == k-> parent -> left) {
          k = k-> parent;
          right Rotate (k);
       }
       k-> parent -> color = 0;
       k-> parent -> parent -> color = 1;
       left Rotate (k-> parent -> parent);
    }
} else {
   u = k-> parent -> parent -> right;
   if (u-> color == 1) {
      u -> color = 0;
      k -> parent -> color = 0.
      k -> parent -> parent -> color = 1;
      k = k-> parent -> parent;
  } else {
    if (k == k-> parent -> right) {
            k = k -> parent ;
            left Rotate (k) ;
        }
      k -> parent -> color = 0;
      k -> parent -> parent -> color = 1;
      rightRotate (k -> parent -> parent);
    }
} if (k == root) {
        break,
    }
} root -> color = 0;
}
void  insert (int key) {
  Node  node = new Node;
    node -> parent = nullptr;
    node -> data = key;
    node -> left = TNULL;
    node -> right = TNULL;
    node -> color = 1;

  Node  y = nullptr;
  Node  x = this -> root;
  while (x != TNULL) {
      y = x;
      if (node -> data < x -> data) {
          x = x -> left;
      } else {
          x = x -> right;
      }
}
```

```
node -> parent = y ;
if (y == nullptr) {
    root = node ;
} elseif (node-> data < y ->data) {
      y -> left = node ;
} else {
   y -> right = node ;
}
 if ( node -> parent == nullptr) {
     node -> color = 0 ;
     return ;
 }
  if (node -> parent -> parent == nullptr)
    {return ; }
  insertfix (node);
 }
```