ADS - LAB

Khrithik . S Anand
1BM18CS046

Insert Function

```
void insert(int key){
    Node node = new Node;
    node->parent = NULL
    node->data = key;
    node->left = NULL
    node->right = NULL
    node->colos = 1
    Node ptr y = NULL
    Node ptr 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) root = node;
        else if(node->data < y->data) y->left = node;
        else y->right = node;
        if(node->parent) node->colos = 0; return;
        if(!node->parent->parent) return;
        insertfix(node);
}
void InsertFix(Node ptr k){
    Node ptr v;
```

ADS - LAB

Kavitha S. Anand
1BM18CS046

```
while (k->parent->color == 1){
    if (k->parent == k->parent->parent->right){
        if (u->color == 1){ u = k->parent->parent->left;
            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;
                rightRotate(k);
            }
            k->parent->color = 0
            k->parent->parent->color = 1
            leftRotate (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;
                leftRotate(k);
            }
```

Prathik S. Anand
IBM18CS041

```
        k->parent->color=0;
        k->parent->parent->color=1;
        Right rotate(k->parent->parent);
      }
    }
    if(k==root) break;
  }
  }
  root->color=0
}
```