AKSHAY MITTUR
1BM18CS010

18/11/20

ADS LAB WRITEUP

Red Black Tree Insertion

Insert Function:

```
void insert (int key) {
        NodePtr node = newNode;
        node->parent= NULL;
        node->data =key;
        node->left= NULL;
        node->right = NULL;
        node->color = 1;
        Nodeptr y= NULL;
        Nodeptr 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->color=0, return;
         if (!node->parent->parent) return;
        insertfix(node);
}

void insertFix(NodePtr k) {
        Nodeptr u;
        while (k->parent->color==1) {
            ~~if (k->parent->color ==1){~~
            if (k->parent == k->parent->parent->right) {
                    u= k->parent->parent->left;
```

AKSHAY MITTOR
1BM18CS010

Page2

18/11/20

```
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;
            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->colour= 1;
        k= k->parent-> parent;
    } else{  if( k==k->parent->right) {
            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;
}
```