14-10-20

ADS-lab

B Praneeth
1BM18CS023

AVL tree

Insertion

```
NODE insert (NODE head, int ele) {
    if ( ~~key~~ ele < head -> value)
        head -> left = insert (head -> left, ~~key~~ ele);
    else if (ele > head -> value)
        head -> right = insert (head -> right, ele);
    else
        return head ;
    balance = getBalance(head);
    if (balance <-1 && val > node -> ~~left~~ right -> value)
        return leftRotate( ~~node~~ head);
    if (balance > 1 && val < head -> left -> value)
        return rightRotate (head);

    if (balance >1 && ele > head -> left -> value){
        head -> left = leftRotate (head -> left);
        return rightRotate( ~~hed~~ head );
    }
    if (balance <-1 && key < ~~node~~ head -> right -> value)
```

14-10-20 ADS-lab B Praneeth
1BM18CS023

```
        head -> right = rightRotate (head -> right);
        return leftRotate (head);


    return head;

}
```

Deleting a node

```
NODE deleteNode (NODE head, int ele)
        if (ele < head -> value)
            head -> left = deleteNode (head -> left, ele);
        else if (ele > head -> value)
            head -> right = deleteNode (head -> right, ele
        ~~root~~ head -> height = 1 + max(height (head -> left),
                                       height(head -> right));
        int balance = getBalance(head);
    if (balance > 1 && getBalance (~~root~~ head -> left) >= 0)
            return righRotate (head);

    ~~// Remaining cases are~~

    if (balance > 1 && getBalance (head -> left) < 0)
        ~~root~~ head -> left = leftRotate (head -> left);
        return righRotate (head);
} /* Remaining cases are same as done during insertion ?
```

B Praneeth