

18-11-2020

ADS lab

B Praneeth
18M18CS023

Red-black tree, insertion

// insert function

void insert(int n) {

Node *newNode = new Node(n);

if (root == NULL) {

newNode → color = BLACK;

root = newNode;

}

else {

Node *temp = search(n);

if (temp → val == n) {

return; // element already exists

newNode → parent = temp;

if (n < temp → val)

temp → left = newNode;

else

temp → right = newNode;

fixRedRed(newNode);

}

}

RB

18-11-20

B Praneeth

18M18CS023

// fix red red violation

```
if (x == root) {
```

```
    x → color = BLACK;
```

```
    return;
```

```
}
```

```
Node *parent = x → parent, *grandparent = parent → parent,  
*uncle = x → uncle();
```

```
if (parent → color != BLACK) {
```

```
    if (uncle != NULL && uncle → color == RED) {
```

```
        parent → color = BLACK;
```

```
        uncle → color = BLACK;
```

```
        grandparent → color = RED;
```

```
        fixRedRed(grandparent);
```

```
    }
```

```
else {
```

```
    // Perform LR, LL, RL, RR rotations
```

```
    if (parent → isOnLeft()) {
```

```
        if (x → isOnLeft()) {
```

```
            swapColors(parent, grandparent);
```

```
        } else {
```

```
            leftRotate(parent);
```

```
            swapColors(x, grandparent);
```

```
        }
```

Red-black tree

// LL and LR

rightRotate(grandparent);

} else {

if (x → isOnLeft()) {

// RL

rightRotate(parent);

swapColors(x, grandparent);

} else {

swapColors(parent, grandparent);

}

// RR and RL

leftRotate(grandparent);

}

}

}

{