18/11/2020

ADS Lab -8

Triveni.y
1BM19CS411

Write a program to implement insertion operation on a red-black tree. During insertion, appropriately show how recolouring or rotation operation is used.

```
void RBTree :: insert (const int &data)
{
    Node *pt = new Node(data);

    root = BSTInsert (root, pt);
    fixViolation (root, pt);
}
```

BSTInsert ():-

```
Node* BSTInsert(Node* root, Node *pt)
{
    if (root == NULL)
        return pt;

    if (pt->data < root->data)
    {
        root->left = BSTInsert(root->left, pt);
        root->left->parent = root;
    }
    else if (pt->data > root->data)
    {
        root->right = BSTInsert(root->right, pt);
        root->right->parent = root;
    }
```


Y. Triveni

```
return root;
}

void RBTree:: fixViolation(Node *&root,
                           Node *&pt)
{
    Node *parent_pt = NULL;
    Node *grand_parent_pt = NULL;

    while((pt != root) && (pt->color != BLACK)
        && (pt->parent->color == RED))
    {
        parent_pt = pt->parent;
        grand_parent_pt = pt->parent->
                          parent;

        if (parent_pt == grand_parent_pt
                         ->left)
        {
            Node *uncle_pt = grand_parent_pt
                             ->right;

            if (uncle_pt != NULL &&
                uncle_pt->color == Red)
            {
                grand_parent_pt->color = Red;
                parent_pt->color = Black;
                uncle_pt->color = Black;
                pt = grand_parent_pt;
```


Y. Trivi

Trivenij

```
else
{
    if(pt == parent_pt -> right)
    {
        rotateLeft(root, parent_pt);
        pt = parent_pt;
        parent_pt = pt -> parent;
    }

    rotateRight(root, grand_parent_pt);
    swap(parent_pt -> color, grand_parent_pt
                -> color);
    pt = parent_pt;
}
}

void RBTree :: rotateLeft (Node *&root, Node
                            *&pt)
{
Node * pt_right = pt -> right;
pt -> right = pt -> right -> left;
if (pt -> right != NULL)
    pt -> right -> parent = pt;
pt_right -> parent = pt -> parent;
```

Triveni.y

Bafna Gold

Date: Page:

```
if (pt -> parent == NULL)
        root = pt->right;
else if (pt == pt->parent->left)
        pt->parent->left = pt->right;
else
    pt->parent->right = pt->right;

pt->right->left = pt;
pt->parent = pt->right;
}
```