AI-LAB-7

1BM18CS065
18-11-2020
Nitish. N. Banakar

B-Tree :-

```
insert (data):
    pt = new Node(data)
    root = BSTInsert(root,pt)
    fixViolation(root, pt)

BSTInsert (root, pt):
    if (root is NULL):
        return pt
    if (pt -> data < root->data)
        root -> left = BST Insert(root->left, pt)
        root -> left -> parent = root
    else if (pt->data > root->data)
        root -> right = BST Insert (root->right, pt)
        root -> right -> parent = root;
    return root

levelOrder Helper (root):
    if (root is NULL)
        return
    queue < Node *> q;
    q. push (root);
    while (!q.empty())
        temp = q.front();
        cout << temp -> data << " ";
```

[illegible]

```
        q.pop()
        if (temp->left != NULL)
             q.push(temp->left);
         if(temp->right != NULL)
              q.push(temp->right);

rotateLeft (root , pt):
     pt_right = pt->right;
     pt->right = pt_right->left;
     if(pt->right != NULL)
            pt->right->parent = pt;
     pt->right->parent = pt->parent;
     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

rotate Right (root , pt)
      pt_left = pt->left;
      pt->left = p_left->right;
      if (pt->left != NULL)
                pt->left->parent = pt;
       pt_left->parent = pt->parent
```

[illegible]

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

fixVoilation(root, pt):
    parent_pt = NULL, 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)
           uncle_pt =grand_parent_pt->right
          if(uncle_pt !=NULL && uncle_pt->color==[illegible]
             grand_parent_pt->color =RED
             parent_pt->color =BLACK;
             uncle_pt->color = BLACK;
             pt = grand_parent_pt;

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

[illegible]

```
rotateRight (root, grand-parent-pt);
swap (parent-pt->color, grand-parent-pt->color);
pt = parent-pt.
```

[illegible]