Vinay Kumar

28/10/20 1BM18CS153 DATE: PAGE:

Lab - 6

2-3 tree

Insertion

```
public void insert(int k) {
    if( root == null)
    {
        root = new BTreeNode(t,true);
        root.keys[0] = k;
        root.n = 1;
    }
    else {
        if( 2*t -1 == root.n)
        {
            BTreeNode newRoot = new BTreeNode(t,false);
            newRoot.C[0] = root;
            newRoot.splitChild(0,root);
            int i = 0;
            if (newRoot.keys[0] < k) {
                i++;
            }
            newRoot.C[i].insertNonFull(k);
            root = newRoot;
        }
        else
            root.insertNonFull(k);
    }
}

public void insertNonFull(int k) {
    int i = n-1;
    if (leaf)
        while( i >= 0 && keys[i] > k)
        {
```

```
keys[i+1] = keys[i]
  i--;
key[i+1] = k;
n = n+1;
}
else {
  while (i>=0 && keys[i] >k)
    i--;
  if ((c[i+1].n == 2*t-1)
    splitChild(i+1, c[i+1]);
    if( keys[i+1] <k)
      i++;
  }
    c[i+1].insertNonFull(k);
  }
}
```

Delete

```
Private boolean remove(Node current, T element){
  boolean ifRemoved = true;
  if(current == null) {
    ifRemoved = false;
    return false;
  }

else {
    if(!current.getLeftElement().equals(element))
    {
      if(current.getRightElement ==NULL ||
        current.getRightElement().compTO(element)
                                  == ROOT-IS-BIGGER)
```

```
if( !current.getRightElement().equals(element)){
   ifRemoved = remove(current.rightchild, element);
}
else {
   if (current.isLeaf()){
      current.setRightElement(NULL);
   }
   else {
      T replacement = (T).current.getRightNode().
                        replacemin();
      current.setRightElement(replacement);
   }
}
}
if (current.getRightNode() != NULL) {
   if((current.getMidNode()).isLeaf() && !current.getRight
                        Node().isLeaf() {
      current.getRightNode.ReBalance();
   }
   if(current.getMidNode().isLeaf() && !current.
         getRightNode().isLeaf(){
      T replacement = (T) current.getrightnode().setmin();
      T tempRight = (T) current.getRightElement();
      current.setRightElement(replacmat);
      add(tempRight);
   }
   else {
         isBalanced = true;
      }
}
return ifRemoved;
}
```