LAB 5 1BM18CS077

Rahul Patil

```
void Tree::insert(int k) {
if(root == NULL) {
root = new TreeNode(true);
root -> keys[0] = k;
root -> n = 1;
 }else {
if(root -> n == 3) {
TreeNode *s = new TreeNode(false);
s -> child[0] = root;
s -> splitChild(0, root);
int i = 0;
if(s -> keys[0] < k) i++;
s -> child[i] -> insertNonFull(k);
root = s;
 }else
root->insertNonFull(k);
 }
}
void TreeNode::remove(int k) {
int idx = findKey(k);
if(idx < n && keys[idx] == k) {
if(leaf) removeFromLeaf(idx);
else removeFromNonLeaf(idx);
 }else {
if(leaf) {
cout << "The key doesn't exist \n";
return;
  }
bool flag = ((idx == n)?true  false);
if(child[idx] -> n < 2) fill(idx);
if(flag && idx > n) child[idx - 1] -> remove(k);
else child[idx] -> remove(k);
 }
return;
}
```