

Arbax Ahmed
1BM19CS401

Q-3 tree insertion

insert (k)

if !root

root = new node // leaf
root->keys[0] = k
root->n = 1

else

if root->n = 3

s = new node

s->child[0] = root

s->splitChild(0, root)

int i = 0

if s->keys[i] < k

s->child[i+1] = insertNonFull(k)

root = s

else

root->insertNonFull(k)

8 \rightarrow insertNonFull(k)

9 $p = n - 1$

10 if leaf

11 where $p > 0$ & $key[p] > k$

12 $key[p+1] = key[n]$

1 $key[p+1] = k$

2 $n = n + 1$

3 else

4 while $p > 0$ & $key[p] > k$

5 $p--$

6 if child $[p+1] \rightarrow n = 3$

7 splitchild $[p+1, child[p+1]]$

8 if $key[p+1] < k$

9 $key[p+1] \rightarrow insertNonFull(k)$

delete(k)

$i = \text{find}(k)$

if $i \leq n$ & $\text{key}[i] = k$

if leaf

removeleaf(i)

else

removeNonleaf(i)

else

if leaf

Print("Not found");

exit;

flag = (P == n)

if $\text{child}(i) \Rightarrow n$ & $\text{full}(i)$

if $i > n$ & flag

$\text{child}(i) \rightarrow \text{remove}(k)$

return