

MD YASEEN AHMED

1BM19CS404

⇒ 2-3 trees insertions:

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 → split child(0, root)

int i = 0

if s → Keys[0] < k

i++

s → child[i] = insertNonfull(k)

root = s

else

root → insertNonfull(k)

→ insertNonFull(k)

$i = n - 1$

if leaf

while $i > 0$ and $Keys[i] > k$

$Keys[i+1] = Keys[i]$

$i--$

$Keys[i+1] = k$

$n = n + 1$

else

while $i > 0$ && $Keys[i] > k$

$i--$

if $child(i+1) > n - 3$

split child $(i+1, child(i+1))$

if $key[i+1] < k$

~~key~~ $i++$

$key[i+1] \Rightarrow insertNonFull(k)$

⇒ delete K

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

if $i < n$ && $\text{Keys}[i] = K$.

if leaf

removeLeaf[i]

else

removeNonleaf[i]

else

if leaf

print ("Not found");

exit;

flag = ($i == n$)

if $\text{child}[i] > n < t$

fill [i]

if $i > n$ && flag

$\text{child}[i-1] \rightarrow \text{remove}(K)$

else

$\text{child}[i] \rightarrow \text{remove}(K)$

return