

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[0] < k
 i++

s → child[i] = insertNonFull(k)

root = s

else

root → insert ^{Non-Full} ~~Non-Full~~ (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

if child(i+1) → n = ~~2~~ 3

```

split child (i+1, child(i+1))
if key(i+1) < k
    i++
    (child(i+1)) → insert Non Full (k)

```

remove k

```

i = Find(k)
if i < n & keys(i) = k
    if leaf
        remove Leaf(i)
    else remove NonLeaf(i)
else
    if leaf
        print (not there)
        exit ; flag = (i == n)
    if child(i) → n < i
        fill(i)
    if i > n & flag
        child(i-1) → remove(k)
    else
        child(i) → remove(k)
return

```

Ans:

remove Leaf → shift elem to left after deletion
 remove NonLeaf → merge nodes after deletion
 split child → split node into children nodes