Name: Rohit Kudache

USN: 1BM18cs083

Insertion In B-Tree – Program 7

→ If tree is empty
then allocate a root node and insert key.

→ Update the allowed number of keys in the node.
→ Search the appropriate node for insertion.

if node is full
→ Insert the elements in increasing order
Now, there are elements greater than it's limit
So split at the median.
→ push the median key upwards and make the left keys as a left child and right keys as a right child.

else (if node is node full)
→ Insert node in increasing order

```
class BTreeNode:
    def __init__(self, leaf=False):
        self.leaf = leaf
        self.keys = []
        self.child = []

class BTree:
    def __init__(self, t):
        self.root = BTreeNode(True)
        self.t = t
```


Rohit

1BM18CS083
RK

Page No.
Date

```
def insert (self, k) :
    root = self.root
    if len (root.keys) == (2 * Self.t) - 1
        temp = BTreeNode ()
        self.root = temp
    temp.child.insert (0, root)
    Self.split_child ~~in sert~~ (temp, 0)
    Self. insert_non_full (temp, k)
else :
    Self. insert_non_full (root, k)
```

02

Insert