ADS - Lab 6

MANU.N.Y
IBM18CS053

B-Tree Insertion

```
insert (int k)
  if (root is NULL)
     root = new node(t);
     root ⟹ keys[0] = k;
      root → n = 1;
  else
    if(root → n == 2*t - 1)
       s = new Node(t);
       s → c[0] = root;
       s → splitchild (0, root);
       i = 0
       if (s → keys[0] < k)
          i++;
       s → c[i] → inseat NonFull(k)
       root = s
    else
      root → inseat Non Full (k)
```

```
Class Node{
  int *keys;
  int t;
  Node **C;
  int n;
  bool leaf;
}
```

```
Node Inseat NonFull (int k)
    i = n - 1
    if (leaf is true)
      while (i >= 0 && keys[i] > k)
        do keys[i+1] = keys[i]
           i--;
      keys[i+1] = k
      n = n + 1
    else
      while (i >= 0 && key[i] > k)
        do i--
      if (c[i+1] → n == 2*t - 1)
        Splitchild (i+1, c[i+1]);
        if (keys[i+1] < k)
        c[i+1] i++; → inseat NonFull(k)
```

```
Node splitchild (int i, Node *y)
  z = new Node (y->t)
  z->n = t-1
  for (int j=0; j<t-1; j++)
    z->key[j] = y->keys[j++]
  if(y->leaf is false)
    for (int j=0; j<t; j++)
      z->C[j] = y->c[j++]
  y->n = t-1
  for (j=n; j>=i+1; j--)
    c[j+1] = c[j]
  c[i+1] = z
  for (j=n-1; j>=1; j--)
    keys[j+1] = keys[j]
  key[i] = y->keys[t-1]
  n = n+1
```