Ajay Mittur
IBM17IS006

Date

Expt. No.

B-Tree

Page No.

```
void insert (int k) {
    if (!root) {
        root = new Node(t, true);
        root->keys[0] = k;
        root->n = 1;
    }
    else if (root->n == 2*t - 1) {
        Node * s = new Node(t, false)
        s->C[0] = root;
        s->split(0, root);
        int i = s->keys[0] < k ? 1 : 0;
        s->C[i]->insertNonFull(k);
        root = s;
    else root->insertNonFull(k)
}
void insertNonFull(int k) {
    int i = n-1;
    if (leaf) {
        while ( i >= 0 && keys[i] > k) {
            keys[i+1] = keys[i];
            i--;
        }
        keys[i+1] = k;
        n = n+1;
    } else {
        while (i >= 0 && keys[i] > k) i--;
```

Teacher's Signature :

B-Tree

```
if (((C[i+1]->n == 2*t-1) {
    split (i+1, (C[i+1]);
    if (keys[i+1] < k) i++;
    (C[i+1] -> insertNonFull (k);
}
}

void split (int i, Node* y) {
    Node * z = new Node (y->t, y->leaf);
    z->n = t-1;
    for ( int j=0; j<t; j++ ) z->keys[j] = y->keys[
                                           j+t];
    if (y->leaf == false) {
      for ( int j=0; j<t; j++ )
          z->C[j] = y->C[j+t];
    }
    y->n = t-1;
    for (int j= n; j >= i+1; j--)
          C[j+1] = C[j]

    C[i+1] = z;
    for (int j=n-1; j>=i; j--)
      keys[j+1] = keys[j];

    keys[i] = y->keys[t-1];
    n = n+1;
}
```