BTREE INSERTION

```
Void BTree :: insert (int k)
                    // if tree is empty
{       if root is empty
        { root = new BTreeNode(t, true);
          root -> key[0] = k;
          root -> n = 1;
        }   // if tree is not empty
else
{       // if root is full   // Splitting happens
        if (root -> n == 2*t - 1)
        {
            BTree * s = new BTreeNode
                        (t, false);
            s -> C[0] = root;
            s -> SplitChild (0, root);

            int i=0;
            if (s -> keys[0] < k)
              i++;
            s -> C[i] -> insertNonFull(k);

            root = s
        }
        else   // if not full
        root -> insertNonFull(k);
}
```

```
void BTreeNode :: insertNonFull(int k)
{
    int i = n-1;   // rightmost element.
    if (leaf == true)  If it is an leaf node.
    {
        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--;
        if (C[i+1] -> n == 2*t-1)
        {   splitChild (i+1, C[i+1]);
            if (keys[i+1] < k)  i++;
        }
        C[i+1] -> insertNonFull(k);
    }
}
```