ADS Lab 7

```
func insert(int k):
    if root == NULL:
        root = new BTreeNode(t);
        root -> key[0] = k;
        root -> n = 1;
    else:
        if (root -> n == 2*t - 1):
            S = new BTreeNode(t)
            s -> c[0] = root
            s -> splitChild(0, root)
            int i = 0;
            if (s -> key[0] < k)  i++
            s -> c[i] -> insertNonFull(k);
            root = s
        else: root -> insertNonFull(k);

func insertNonFull(int k):
    i = n-1
    if leaf == true:
        while (i >= 0 and keys[i] > k):
            keys[i+1] = keys[i]; i--
        keys[i+1] = k ; n += 1
    else:
        while (i >= 0, and 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)
```

```
func  splitChild(int i, BTreeNode y):
    z = new BTreeNode(y->t, y->leaf)
    z->n = t-1
    for i from 0 -> t-1:
        z->key[i] = y->c[j+t];
    y->n = t-1
    for j from i+1 -> n:
        c[j+1] = c[j]
    c[j+1] = z
    for j from n-1 -> i:
        keys[j+1] = keys[j]
    keys[i] = y->key[t-1]
    n = n+1;
```