Program 7

Naman Agarwal
1BM18CS057

```
class Node
{
    int * keys;
    int min;
    Node * * child;
    int number_of_keys;
    bool isleaf;

    @constructor ( t, leaf)
        isleaf = leaf
        min = t.
        keys = new int [2*t - 1]
        child = new *Node [2*t]
        number_of_keys = 0.

    insert if not full ( int key)
        if (isleaf = true)
            // find the appropriate place
                in the correct node
        else
        {
            while ( i >= 0 & key [i] > k)
            {
                if (child [i+i] -> n = = 2*t-1)
                    split
                if (key [i+1] < k)
                    i++.
                child (i+1) -> insert not full
            }
```

N. Agarwal

SMILE | its Infectious

Program-8

Naman Agarwal
1BM18CS057

split (i, node *temp)

node *z → new node.

z → n = min - 1

Copy last (t-1) key of temp to z

if (y → isleaf = false.

copy last (t) children.

link new child → ch[i+1] = z.

keys[i] = y → keys[t-1]

number of keys ++.

Class B-tree

$ Node * root.

@ constructor.
Initialize root

insert

if root = Null
↳ put it in root node.

else traverse in the $ place.

N. Agarwal

SMILE ... ITS INFECTIOUS