Suraj Perdoor IBM18CS114

ADJ LAB 6   b-Tree

```
def insertion (7, h):
    root = self.root

    if (len (root.keys) == (2 * self.t) - 1:

        temp = Node ()
        self.root = temp
        temp.child.insert (0, root)
        self.split_child (temp, 0)
        self.insert_nonfull (temp, h)
    else:
        self.insert_nonfull (root, h)


def insert_nonfull (self, n, h)

        i = len (n.keys) - 1
        if (n.leaf):
            n.keys[i+1] = n.keys[i]

            while (i >= 0) and h[0] < n.keys[i][0]:
                n.keys[i+1] = n.keys[i]
                i -= 1
            n.keys[i+1] = h

        else:
            while i >= 0 and h[0] n.keys[i][x-0] =
                i -= 1

            i += 1
            if (len (n.child.keys) == (2 * self.t) -1 =
                self.split_child (n, i)
                if (h[0]) > n.keys[i][0]:
            self.insert_nonfull (n.child[i], h)
```