Program-6

Naman Agarwal
IBM18CS057

Insertion

insert (node, compare value)
. locate a leaf to put value in it
if leaf is a 2 node, make it a 3 node
inserting the value appropriately.
if leaf is a 3 node, split the node.

split (node N)
if N is root
make middle child int a 2 node
make small & large key into 2 node.
Reduce children.

else N has a parent P → move middle key to P.
make small and large children into a 2 node
Rending children.

insert (a, u)
if (u consists of a single leaf value b)
create a node u.
create a new leaf v
make l and v children of u.
update u'.
else
set f to search (a, T)
create a new leaf l called a.
If f has 2 children

A. Agarwal

Naman Agarwal
IBM18CS057

insert a into proper position,
update L and h.

else
create a transitory node tree at f.
and child(f)

}

Addchild (v)

{

create new node v'
move 2 rightmost children of v to v'
if (v has no parent)
make new nat u'
make v left child and v' right child.
update h-f h.
else
let f be parent of v.
make v' child of f immidiately to right if f now has 4 children.
Addchild (f)
else
update h & u

}

N. Agarwal.

Naman Agarwal
1BM18CS057

```
delete ()
{
    let f be parent of node just deleted
    while ( f is an illegal internal node)
    {
        if (f has no parent)
            make single child of f new node
            delete f
            set f to the root

        else
            let g be parent of f.
            if (one of f's sibling is a 3 node)
                move one child (from 3-node into
                f, update k1 & k2 everywhere

            else
                give f's remaining child to one
                f's sibling
                delete f
                update k1 & k2 everywhere.
                set f to g.

    }

}
```