(1) Advanced Data Structures Lab

Batch - 5

Program - 5

Rutazeet Ritik Rout
1BM18CS151

28/10/20

Insertion :-

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

Split (node N)
if N is root
make middle child into 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 2
2 node
Reducing children.

insert (a, r){
if (r consists of a single leaf labeled b)
(reate a new ~~node~~ root r'
(reate a new leaf v labeled a
make l & v children of r'
update L & M for r'

else
Set f to search (a, r)
(reate a new leaf l labeled a

Rutazeet.

(2)

```
if f has 2 children
    insert l into proper position
    update L & H
else
    create a transitory node tree at f
    Addchild (f)
}
```

```
Addchild (v) {
    create new node v'
    move 2 right most children of v to v'
    if (v has no parent)
        make new root r'
        make v: left child & v' : right child
        update L & H
    else
        let f be parent of v
        make v' child of f immediately to right of v
        if f now has M children
        Addchild (f)
        else
            update L and H.
}
```

(3)

```
delete ( ){
    let f be parent of node just deleted
    while( f is an illegal interior node) {
        if ( f has no parent)
            make single child of f. new node
            delete f.
            set of 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
    }  }
}
```

Rutuja.