ARAVINDH R
IBM18CS145

ADS

```
func insert (value):
    node current = header;
    node update = [MAX_LVL +1]

    for i from level -> 0:
        current = current.forward[i] if current
    update[i] = current
    current = current.forward[0]

if !current or current.key != key:
    rlevel = random_level()
    if rlevel > level:
        for i from level -> rlevel:
    level = rlevel

node_n = Node(rlevel, key)

for i from 0->rlevel:
    node_n.forward[i] = update[i].forward[i]
    update[i].forward = n
```

```
func delete ( key):
    update = [None] * MAX_LVL + 1
    current = header

    for i from level → 0:
        current = current.forward[i]
    update[i] = current
    current = current.forward[0]

if ! current or current.key != key:
    for i from 0 → level+1
        if update[i].forward[i] != current:
            break
        update[i].forward = current.forward[i]

    while level > 0 and !header.forward[level]:
        level -= 1
```

```
def search(key):
    current = header

    for i from level → 0:
        while current and current.key < key:
            current = current.forward[i]

    current = current.forward[0]

    if current and current.key == key:
        return key.
```