

* Write-up

1) Create a SkipListNode containing 3 data fields:-

- (i) Value
- (ii) Level
- (iii) SkipListNode next[] array
(pointers that store next nodes' address)

2) Create head pointers that points to maximum-level in SkipList and current_max_level variable to tell us what is present max-level.
(In the beginning (for one node). current_max_level = 1).

3) Now while, inserting(value):

```
level = tossing();
currp = head;
for (int i = current_max_level; i >= 0; i--) {
    while (currp.next[i] != null) {
        if (currp.value > value)
            break;
        currp = currp.next[i];
    }
    if (i <= level) {
        newnode.next[i] = currp.next[i];
        currp.next[i] = newnode;
    }
}
```

```
tossing()
{
    head = true;
    level = 0;
    for (i = 0 to max_level)
        head = head & randomBoolean;
    if (head)
        level++;
    return level;
}
```

// Tossing helps us determine the level of each node by probability. [Referenced from Web]

4.) Contains (value): {

currp = head

for (int i = curr-max-level; i >= 0; i--) {

while (currp.next[i] != null) {

if (currp.next[i].value > value) {

break; }

if (currp.next[i].value == value) {

return true; }

currp = currp.next[i];

}

} return false;

}

5.) delete (value): {

currp = head;

for (int i = curr-max-level; i >= 0; i--) {

while (currp.next[i] != null) {

if (currp.next[i].value > value) {

break; }

if (equals) {

currp.next[i] = currp.next[i].next[i];

break; }

currp = currp.next[i];

}

}

} ~~return~~

// Because of curr-max-level variable, we need not do any manipulations to heads (levels).