

Skip lists pseudo codes

The level for node is decided by the following algorithm –

```
randomLevel()
lvl := 1
//random() that returns a random value in [0...1)
while random() < p and lvl < MaxLevel do
  lvl := lvl + 1
return lvl
```

Insertion:

```
Insert(list, searchKey)
local update[0...MaxLevel+1]
x := list -> header
for i := list -> level downto 0 do
  while x -> forward[i] -> key < searchKey do
    x := x -> forward[i]
  update[i] := x
x := x -> forward[0]
lvl := randomLevel()
if lvl > list -> level then
  for i := list -> level + 1 to lvl do
    update[i] := list -> header
  list -> level := lvl
x := makeNode(lvl, searchKey, value)
for i := 0 to list -> level do
  x -> forward[i] := update[i] -> forward[i]
  update[i] -> forward[i] := x
```

Searching:

```
Search(list, searchKey)
x := list -> header
-- loop invariant: x -> key < searchKey
while x -> forward[0] -> key < searchKey do
  x := x -> forward[0]
if x -> key = searchKey then return x -> value
else return failure
```

Deletion:

```
Delete(list, searchKey)
local update[0..MaxLevel+1]
x := list -> header
for i := list -> level downto 0 do
  while x -> forward[i] -> key < searchKey do
    x := x -> forward[i]
  update[i] := x
```

```
x := x -> forward[0]
if x -> key = searchKey then
  for i := 0 to list -> level do
    if update[i] -> forward[i] ≠ x then break
    update[i] -> forward[i] := x -> forward[i]
  free(x)
  while list -> level > 0 and list -> header -> forward[list ->
level] = NIL do
    list -> level := list -> level - 1
```

APV