Skip list

Krishna prasad V
1BM19CS047
30.9.20

```
skiplist (int maxlvl, float p)
{   level = 0
    header = new Node (-1, maxlvl)
}
randomlevel ()
{   r = ((float) rand()) / rand_max
    lvl = 0
    while (r < p && lvl < maxlvl)
    {   lvl++
        r = ((float) rand()) / rand_max
    }
    return lvl
}
insertelement (int key)
{   Node *curr = header
    Node * upd [maxlvl+1]
    memset (upd, 0, sizeof (Node) * (maxlvl+1))
    for (int i = level; i >= 0; i--)
    {   while (curr -> fwd [i] != NULL && curr -> fwd[i] -> key < key)
            curr = curr -> fwd [i]
        upd [i] = curr
    }
    curr = curr -> fwd [0]
```

```
if (cur == NULL || cur->key != key)
{
    nlvl = randomLevel()
    if (nlvl > level)
    {
        for (i = level+1; i < nlvl+1; i++)
            upd[i] = header
        level = nlvl
    }
    Node *n = createNode(key, nlvl)
    for (i=0; i<nlvl; i++)
    {
        n->fwd[i] = upd[i]->fwd[i]
        upd[i]->fwd[i] = n
    }
    cout << print(key)
}
}
deleteElement(key)
{
  Node *cur = header
  Node *upd[maxlvl+1]
  memset(upd, 0, sizeof(Node*)*(maxlvl+1))
  for (i=level; i>=0; i--)
  {
      while (cur->fwd[i] != NULL && cur->fwd[i]->key < key)
          cur = cur->fwd[i]
      upd[i] = cur
  }
  if (cur != NULL && cur->key == key)
  {
      for (i=0; i<level; i++)
          if (upd[i]->fwd[i] != cur)
              break
          upd[i]->fwd[i] = cur->fwd[i]
```

```
        while ( lvl>0 && head->fwd[lvl] == 0)
            lvl--;
        print(key);
    }
}
search (key)
{   Node * cur = head;
    for ( i=lvl ; i>=0 ; i--)
    { while (cur->fwd[i] && cur->fwd[i]->key<key)
        cur = cur->fwd[i];
    }
    cur = cur->fwd[0];
    if ( cur && cur->key == key)
        print(key);
}
```