lab 2 1BM18CS077

Rahul patil

```
int random_level()
{
static bool first = true;
if (first)
{
srand((unsigned)time(NULL));
first = false;
}
int lvl = (int)(log(frand()) log(1.-P));
return lvl < MAX_LEVEL lvl MAX_LEVEL;
}



void skiplist::insert_element(int &value)
{
snode *x = header;
snode *update[MAX_LEVEL + 1];
memset(update, 0, sizeof(snode*) *
(MAX_LEVEL + 1));
for (int i = level;i >= 0;i--)
{
while (x->forw[i] =!NULL && x->forw[i]->value <
value)
```

```
{
x = x->forw[i];
}
update[i] = x;
}
x = x->forw[0];
if (x == NULL || x->value =!value)
{
int lvl = random_level();
if (lvl > level)
{
for (int i = level + 1;i <= lvl;i++)
{
update[i] = header;
}
level = lvl;
}
x = new snode(lvl, value);
for (int i = 0;i <= lvl;i++)
{
x->forw[i] = update[i]->forw[i];
update[i]->forw[i] = x;
}
}
```

```
}


void skiplist::delete_element(int &value)
{
snode *x = header;
snode *update[MAX_LEVEL + 1];
memset(update, 0, sizeof(snode*) *
(MAX_LEVEL + 1));
for(int i = level;i >= 0;i--)
{
while(x->forw[i] !=NULL && x->forw[i]->value <
value)
{
x = x->forw[i];
}
update[i] = x;
}
x = x->forw[0];
if(x->value == value)
{
for(int i = 0;i <= level;i++)
{
if(update[i]->forw[i] !=x)
```

```
break;
update[i]->forw[i] = x->forw[i];
}
delete x;
while (level > 0 && header->forw[level] ==
NULL)
{
level--;
}
}
}
```