Ch - 17

1. a) Aggregate method:-

→ Initially the table has size 1
→ After first insertion, size is 2
→ After second insertion, size is 4
→ After third insertion, size is 8
→ The doubling continues till it reaches size 'n'.

Now considering the cost of n elements,
I st insertion, cost = 1
II nd insertion, cost = 2
[1 for ~~intersection~~ insertion & 1 for copying]
III rd insertion, cost = 4
[1 for insertion, 3 for copying]
IV th insertion, cost = 8
[1 for insertion, 7 for copying]

Total cost of inserting n elements,
$\Rightarrow 1 + 2 + 4 + 8 + \dots + 2^{\log(2n-1)}$

Since it is a geometric series with a common ratio of 2

$$2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^{(\log(2(n)-1))}$$

we get $2^{\log 2(n)-1} \Rightarrow n-1$

~~cost~~ total cost = $O(n-1)$

The amortized runtime for inserting n elements is $O(1)$

b) Accounting method

In this method, cost includes both actual & potential cost.

We will use the potential function

$\phi(T) = 2 * T.num$

Initially the table has size 1 ($T.size = 1$)

$\Rightarrow$ potential in $\phi(T) = 2 * T.num = 0$

For inserting 1 element,

actual cost = 1

potential cost = 2

$\Rightarrow$ amoritized cost = $1 + 2 = 3$

The total amortized cost for inserting n elements is $n * 3$, & the amortized cost per insertion is

$(n*3)/n = 3 \Rightarrow O(3)$

$\therefore$ The amortized runtime for inserting 'n' element $= O(1)$