CH-17

1) Inserting n elements using

a) aggregate method

The table doubles in size when it runs out of space. so if the original size is 1, after insertion it doubles the size to 2 after 2 more insertions it doubles to size 4 etc.

in general after k doublings the size is 2^k

Pseudo code:-

```
initialize table with capacity = 1
    for i = 1 to n
        if table is full:
            newtable = create newtable with si
                       2 * current size.
        copy elements then from old table to
            new table
        table - new table
        insert element i into table
        let, k = log(n+1) - 1
            Total cost = O(n) k
                       = O(n log n)

        cost per insertion = O(log n)
Runtime per insertion is O(log n)

Total time is O(n) * log(n+1)
```

) 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 ($T.size = 1$)

$\therefore$ Potential in $\phi(T) = 2 * T.num = 0$

For inserting 1 element

actual cost = 1

Potential cost = 2

amortized cost = 1+2 = 3

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

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

The amortized runtime for inserting 'n' element

$= O(1)$