

## 1) a) Aggregate method:-

In the aggregate method, we amortize the cost of a series of operations by evenly distributing it among all the operations regardless of their actual cost.

Insertion operation:-

- When inserting an element, if table is not full the operation takes  $O(1)$  time
- When the table is full, needs to be resized which takes  $O(n)$  time to copy the elements to double the size.

$$C_i = \begin{cases} 1 & \text{if table is not full} \\ 2n & \text{if table needs resizing} \end{cases}$$

Number of Insert operations we perform  $n$  operations

Total cost

$$T(n) = \sum_{i=1}^n C_i = \sum_{i=1}^n \begin{cases} 1 & \text{if table not full} \\ 2n & \text{if table is full and needs resizing} \end{cases}$$

Amortized Cost Per operation

$$A(n) = \frac{T(n)}{n} = \frac{1}{n} \sum_{i=1}^n \begin{cases} 1 & \text{if table not full} \\ 2n & \text{if table is full and needs resizing} \end{cases}$$

$$= \frac{1}{n} (n + 2n + \dots + 2n)$$

$$= \frac{1}{n} \times n \frac{(n+1)}{2} \times 2$$

$$= 2n + 3$$

∴ Am Runtime for  $n$  elements =  $O(n)$

## b) Accounting method:-

Insertion

When Insertion element if table not full, charge = 2 to cover future resizing.

When table is full, It needs to be resized, but the extra charges assigned during previous Insertions.

$$a_i = \begin{cases} 2 & \text{if table is not full} \\ 0 & \text{if table is full and needs resizing} \end{cases}$$

Number of Insert operations  
we Perform n Insertions

Total charges :

$$A(n) = \sum_{i=1}^n a_i = \sum_{i=1}^n \begin{cases} 2 & \text{if table is not full} \\ 0 & \text{if table is full and needs resizing} \end{cases}$$

Amortized Cost per operation.

$$\frac{A(n)}{n} = 2$$

A.M runtime for n elements =  $O(1)$ .