HW-17

Aggregate Method :-

① Cost of Expansion : For each expansion, the cost is 2^i

② Total cost of n insertions : The sum of costs for all expansions up to n

Given that the array doubles in size when it needs more space, the total cost of n insertions can be calculated as

$$\sum_{i=0}^{\log_2 n} 2^i = 2^{\log_2 n+1} - 1 = 2n - 1$$

Amortized cost per insertion is :

$$\frac{2n-1}{n} = 2 - \frac{1}{n}$$

As n approaches infinity, the amortized cost approaches 2.

Accounting method :-

① Cost assigned : Assign a cost of 3 to each insertion

② Credit for Expansion : Each expansion gains a credit of $2^i - 1$, where i is the number of insertion since the last expansion.

For each insertion operation, there are two cases

* If no expansion is needed, the actual cost is 1 and the credit is 2.
* If an expansion is needed, the actual cost is 3 and the credit is 2.

The total cost of n insertions can be calculated using accounting method as

$$n \times \text{actual cost} - \text{total credit} = 3n - \sum_{i=0}^{\log_2 n} (2^i - 1)$$

$$= 3n - (2n - 1) = n + 1$$

$\therefore$ The amortized cost per insertion is

$$\frac{n+1}{n} = 1 + \frac{1}{n}$$

As n approaches infinity, the amortized cost approaches 1.

Conclusion:-

Aggregate Method: Amortized cost per insertion is approximately 2.

Accounting Method: Amortized cost per insertion is approximately 1.

The Amortized cost inserting n elements in a dynamic array that doubles in size when it needs more space is bounded and efficient.