Q1) Given a dynamic table that doubles in size when it needs more space. Find the amortized runtime for inserting n elements.

a) Use the aggregate method

b) Use the accounting method

Sol) 1) By Aggregate Method:

a) For the amortized runtime analysis of inserting n elements into a dynamic array that doubles in size.

1) Insertion cost:

i) Each insertion typically costs $O(1)$

ii) When the array is full, resizing occurs, costing $O(n)$ to copy existing elements.

2) Total Resizing cost:

In the form of geometric series

$$1 + 2 + 4 + 8 + \dots + \frac{n}{2} = n-1$$

The total cost $T(n)$ for n insertion is

$$T(n) = O(n) + O(n) = O(n)$$

3. Amortized costs per instruction ...

The Amortized cost per insertion is ...

$$\text{Amortized cost} = \frac{T(n)}{n} = \frac{O(n)}{n} \approx O(1)$$

b) By Accounting Method:

The amortized runtime of inserting n elements into dynamic array using the chaining method

1) Cost setup:

i) Assign an amortized cost of 3 units of each insertion.

ii) Each insertion uses 1 unit for insertion itself and stores 2 units as 'credit' for future resizing.

2) Cost Breakdown:

Regular Insertion: → Costs 1 unit & leaving 2 units in credit. Insertion with Resizing → Array doubles in size, we have enough saved credit to cover the resizing cost.

3. Amortized Cost per Insertion :

For 3 units $= O(1)$, means each insertion has a constant time cost on average, even with occasional resizing.