Ch-17 · HW

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

a) Use the aggregate method.

$$c_i = \begin{cases} \text{cost for insertion with expansion} = i \\ \text{cost for insertion without expansion} = i-1 \end{cases} \quad \text{when } i = \text{power of } 2$$

$$\text{Total cost}, \; T(n) = \sum_{i=1}^{n} c_i \leq n + \sum_{j=0}^{\log n} 2^j$$

$$\leq n + 2n$$

$$= 3n$$

$$\text{Am cost} = \frac{\text{Total cost}}{n}$$

$$= 3n/n = 3$$

$$\text{Am} = O(1)$$

b) Accounting Method

	Actual	Am
Insertion without expansion :	1	3
Insertion with expansion :	$(i-1)$	3

Insertion without expansion : Bank is credited $2

ie; $3-1=2$

Insertion with expansion : Bank : $3-(i-1) = 4-i$

Expansion happen only when i = power of 2

∴ for each expansion, $\frac{i-1}{2}$ insertions

$$\text{Total cost} = \left(i-1-\frac{i-1}{2}\right) \times 2 + (4-i)$$

$$= 2\left(\frac{i-1}{2}\right) + (4-i)$$

$= i - 1 + 4 - i$

$= 3$

$\therefore$ The AM cost for insertion is 3

Since bank balance ≥ 0 at all times

$\therefore$ AM $= O(1)$

The amortised cost of inserting n elements in a dynamic array that doubles in size when it needs more space is 3. This is verified with both the aggregate method and the accounting method.