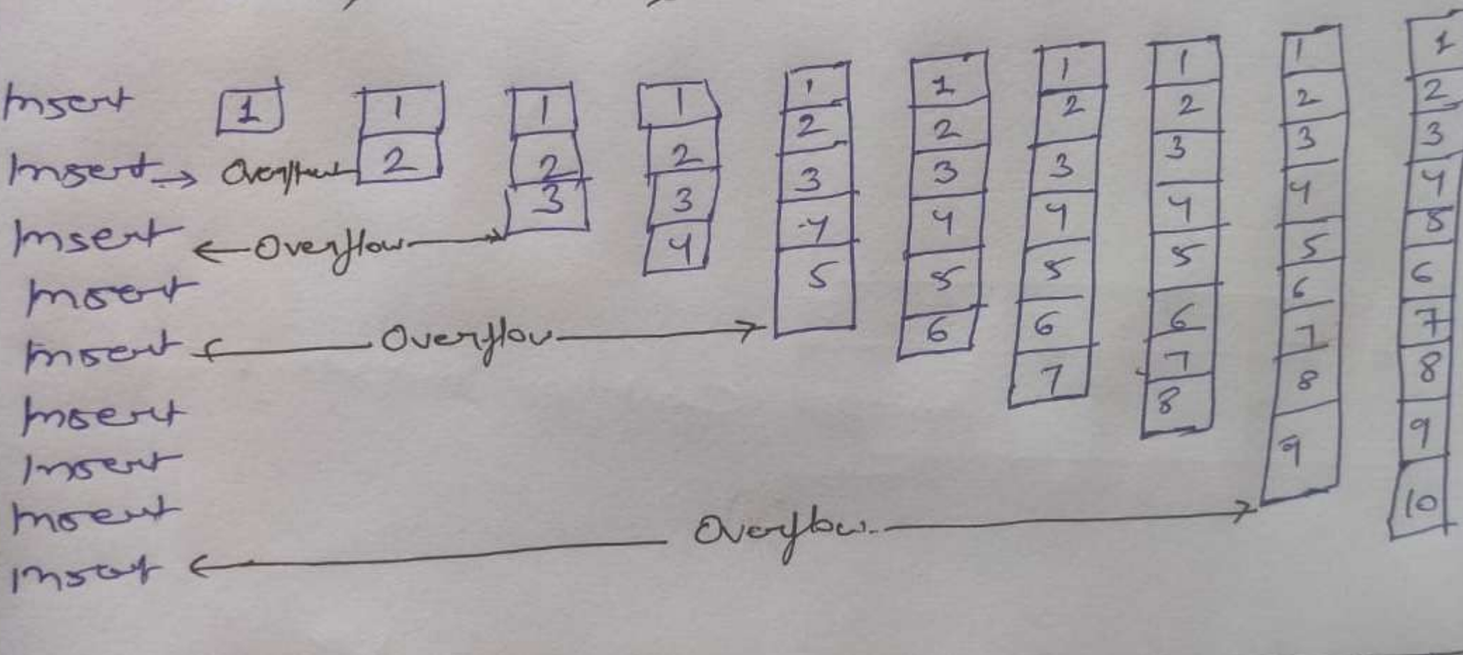
H.W. 17

1 Aggregate method :-

To compute run time of Inserting n elements in dynamic table by Amortized time complexity is computed by aggregate method by the following :-

Eg:- Dynamic Tables



i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Size	1	2	4	4	8	8	8	8	16	16	16	16	16	16	16	16	32
Cost	1	2	3	1	5	1	1	1	9	1	1	1	1	1	1	1	17

$$\text{Cost of } i^{th} \text{ Insertion} = \begin{cases} i, & \text{when } i-1 \text{ as exact power of } 2 \\ 1, & \text{otherwise} \end{cases}$$

$$= n + \sum_{j=0}^{j=\lg(n-1)} 2^j$$

$$= n + (2^0 + 2^1 + 2^2 + --- 2^{\log(n-1)}$$

$$= n + 1\frac{(2^{\log(n-1)+1} - 1)}{1}$$

$$= n + 2^{\log_2(n-1) + \log_2 2} - 1$$

$$= n + 2^{\log_2 2(n-1)} - 1$$

$$= n + 2n - 2 - 1$$

$$= 3n - 3$$

$$\simeq 3n$$

$$\simeq O(n)$$

② Accounting method:-

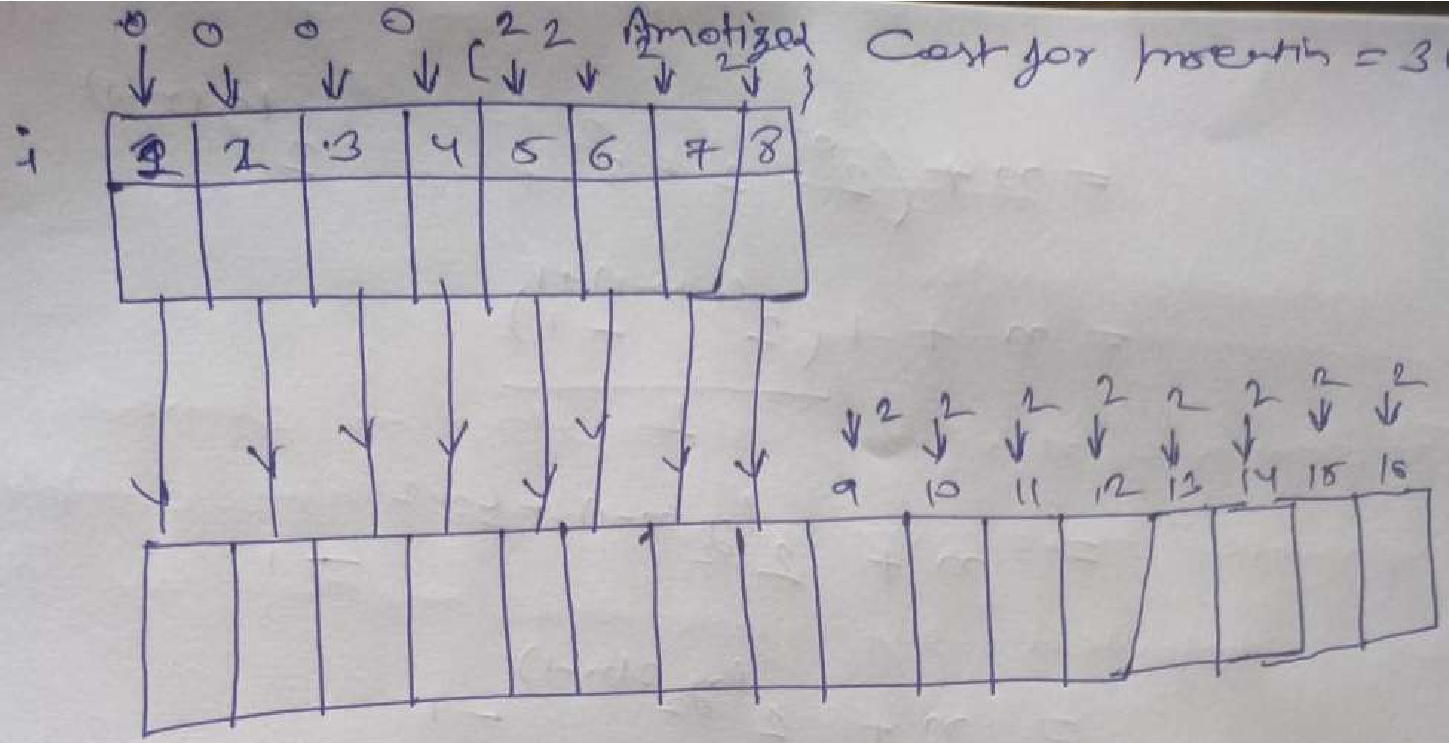
By computing the amortized complexity of dynamic table by accounting method has the following computation :-

In Accounting method:-

The bank balance must never be negatie

means $\underbrace{\sum_{i=1}^{n} c_i}_{\text{Actual cost of no}} \le \underbrace{\sum_{i=1}^{n} \hat{c}_i}_{\text{Amoritze cost assigned.}}$

Amotized Cost for Insertin = 3 Unit



When ever doubling the size in dynamic array, coping each element from old size array element to new size element requires the the cost. It is achieved by saved cost in the past open. Like above example we tryed coping 8 size arry to 16 size arry which require 8 unit of work for coping the data, which was saved in previous array insert operation.

⇒ 8th Unit of time will be used for 9th Insertion

⇒ 16th Unit of time will be used for 17th Insertion

$$\boxed{\begin{aligned} \text{T.C. of Insertion} &= 3*n = O(3n) \\ &= O(n) \end{aligned}}$$