

## Question - 2.

ch - 17.

1002166889

1,

There are two methods we will discuss for inserting  $n$  elements

a, Aggregate method.

→ when the last element is added array doubles its size.

→ for instance: if the array is originally of size 1 that will become 2 and then 4, when it runs out of ~~memory~~ memory.

∴ after  $k$  insertion size will be  $2^k$ .

- pseudocode.

initialize table capacity  $\hat{=}$  1.

for  $i = 1$  to  $n$ :

if table is full.

new table = ~~create~~<sub>2</sub>(new table)



table = new table.

insert element  $i$  into table.

Let  $k = \log(n+1) - 1$ .

$$\begin{aligned}\text{cost} &= O(n) * k \\ &= (n \cdot \log n)\end{aligned}$$

~~Amortized~~ deprecated cost per insertion =  $O(\log n)$

Running time per insertion =  $O(\log n)$

total time is  $O(n) * \log(n+1)$

\* Accounting method.

- ~~for~~ for this method we only double the table after running out of space.
- for copying the table cost will be  $O(n)$

$$\begin{aligned}\text{total credit is } & n + 2n + 4n + \dots + \frac{n}{2} * n \\ & = O(n)\end{aligned}$$



\* pseudo code.

→ initialize the table with capacity = 1.

for  $i=1$  to  $n$ :

if table is full:

newtable = create new table with  
size  $2^+ \text{ curr size}$ .

copy element from old table  
to new table

table = new table.

insert element  $i$  into table.

initialize charges = 0.

initialize credits = 0.

for  $i=1$  to  $n$ :

charges  $t = \alpha$ .

if table doubled in size from  $m$  to  $2m$

credit  $t = n$ .



$$\text{Total charges} = 2n = O(n)$$

$$\text{Total credits} = n + 2n + \dots + n/2 * n = O(n)$$

$$\begin{aligned}\text{Amortized cost per insertion} &= \text{total lu.} \\ &= O(n) / n \\ &= O(1)\end{aligned}$$

$$\text{Runtime per insertion} = O(1)$$

$$\text{Total time} = O(n)$$