

Name : NISARA KACHHIA

UTA ID : 1002265218

DMA @ H.W - 17

①

Inserting  $n$  elements using

a) Aggregate method

→ The table doubles in size when it runs out of space.

→ So, if the original size is  $s$ , after insertion it doubles to size  $2s$  etc

→ In general, after  $k$  doubling the size is  $2^k$

Pseudo code

→ initialise table with capacity = 1  
for  $i = 1$  to  $n$ :

if table is full:

new table = create new table

with size 2 to current size copy element from old to the new table.

table = new table

insert element  $i$  into table

let  $k = \log(n+1) - 1$

$$\text{Total cost} = O(n) \times K$$

$$= O(n \log n)$$

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

Runtime per insertion is  $O(\log n)$

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

(b) Accounting method

\* charge 2 units per each insertion

→ when the table double in size from

$m$  to  $2m$  credit  $m$  units.

→ the credit exactly pay for copy cost at  $O(n)$

→ total credit is  $m + 2m + 4m + \dots + \frac{n}{2}m = O(n)$

Pseudo code?

→ initialise table with capacity = 1  
for  $i = 0$  to  $n$ :

if table is full:

newtable = create with size

2 in current size

copy element from 1 table

to new table.

table = newtable

insert element  $i$  into table

initialize charges = 0

initialize credits = 0

for  $i = 1$  to  $n$ :

charges  $+= 2$

if table doubled in size

from  $M$  to  $< M$ :

credits  $+= M$

total charges =  $2 + 0 + n = O(n)$

total credits =  $M + 2M + \dots + \frac{n}{2}M = O(n)$

A amortized cost per insertion =

total /  $n$ ,  $O(n) / n$

$= 1$

Run time per insertion =  $O(1)$

total time =  $O(n)$