

H.W 17

① (a) aggregate method

Given: Table doubles in size when it needs more space & after  $k$  doublings, the size is  $2^k$ .

Pseudocode:

Initialize table with capacity = 1

for  $i = 1$  to  $n$ :

if table is full:

new table = create new table with size  $2 \times$  current size  
copy elements from old table to new table

table = new table.

insert element  $i$  into table.

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

Total cost =  $O(n) + k$   
 $= O(n \log n)$

A motivated cost per insertion  $= O(\log n)$

Run time per insertion is  $O(\log n)$

$\therefore$  Total time is  $O(n) \times \log(n+1)$

## (b) Accounting Method

(2)

charge 2 units for each insertion

when table doubles in size from  $m$  to  $2m$ ,  
credit  $m$  units.

Total credit is  $m + 2m + 4m + \dots + n/2 * m = O(n)$

### Pseudocode

initialize table with capacity = 1

for  $i = 1$  to  $n$

if table is full:

newtable -- create newtable with size  $2 * \text{current size}$

copy elements from old table to new table

table = newtable

insert element  $i$  into table

initialize charges = 0

credits = 0

for  $i = 1$  to  $n$

charges  $++ 2$

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

credits  $++ m$

Total charges =  $2 * n = O(n)$

credits =  $m + 2m + \dots + n/2 * m = O(n)$

Amortized cost per insertion = Total /  $n$

=  $O(n) / n$

Runtime per insertion =  $O(1)$

total time =  $O(n)$