

## Hands on 11

- ① Given a dynamic table (see section 17.4) that doubles in size when it needs more space. Find the amortized runtime for inserting  $n$  elements.

(a) Use the aggregate method we consider the total cost across all the insertions & calculate the average (amortized) cost per insertion.

When inserting the  $i$ th element if a resize operation is not needed the existing happens cost  $O(1)$  as it involves copying the existing elements to the new table of size  $2^k$  ( $k$  is number of resizes performed).

Accounting method: In this method we assign each insertion a higher "amortized" cost the state "credits" that pay for future resizing costs.

Pseudo Code:

```
For  $i = 1$  to  $n$ 
  if table is full
    new table = create new table
    with size then copy elements from
    old table to new table
    table = new table
  Insert element  $i$  into table
  initial charge = 0
  for  $i = 1$  to  $n$ 
    charges  $t = 2$ 
  if table doubled in size from  $m$  to  $2m$ 
    credits  $t = m$ 
  total charges =  $2 \times n = O(n)$ 
```

$$\text{total credit} = m + 2m + \dots + n/2 \times m = O(n)$$

Amortized cost per insertion

$$= \text{total} / n$$

$$= O(n/n)$$

$$= O(1)$$

Runtime per insertion  $O(1)$

total time  $O(n)$

A growing method: in this method we assign each insertion a higher "amortized" cost the state "credit" that pay for future resizing cost.

Pseudo code:

For  $i = 1$  to  $n$

if table is full

new table = create new table

write size then copy element from old table to new table

table = new table

insert element i into table

initial charge = 0

for  $i = 1$  to  $n$

charge  $t = 1$

if table doubled in size from  $i-1$  to  $i$

credit from

total charge =  $2 \times n = O(n)$