Name : Karan Thakkar DAA HW 17
UTA ID : 1002240528

① Inserting n elements using

a) Aggregate method

→ The table doubles in size when it runs out of space.
→ So, if the Orginal size is 1, after insertion it doubles to size 2, after 2 more insertion it doubles to size 4 etc
→ In general, after k doublings the size is 2^k

pseudo code

→ initialize 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 its new table.
table = new table
insert element i into table
let $k = \log(n+1) - 1$
Total cost = $O(n) * K$
$= O(n \log n)$
Amortize cost per insertion = $O(\log n)$

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

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

(b) Accounting method

* Charge 2 units per each insertion
-> when the table doubles in size from m to 2m, credit m units
-> The credit exactly pay per copy cost at O(m)
-> Total credit is $m + 2m + 4m + \ldots \frac{n}{2} m = O(n)$

pseudo code :

```
-> intialize table with capacity = 1
   for i = 0 to n:
      if table is full:
         newtable = create new table with size
                    2 in current size
         copy elements from old table to new
         table
      table = newtable
      insert element i into table
   intialize charges = 0
   intialize 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)
   Total credits = m + 2m + ... + n/2 m = O(n)
Amortized cost per insertion = Total / n . O(n)/n
                             = 1 //

Runtime per insertion = O(1)
Total time = O(n)  =
```