

DAA

Homework ch-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 1, after insertion it doubles the size to 2 after 2 more insertions 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 $2 \times$ current size

copy elements then from old table to new table

table = newTable

insert element x into table

$$\text{let, } k = \log(n+1) - 1$$

$$\text{Total cost} = O(n) * k$$

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

$$\text{cost per insertion} = O(\log n)$$

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

$$\text{Total time is } \underline{O(n) * \log(n+1)}$$

b) accounting method

charge 2 units for each insertion

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

The credit exactly pay for the
copy cost of $O(m)$.

$$\text{Total credit is } m + 2m + 4m + \dots$$

$$n/2 * m = O(n)$$

Pseudo code:

Initialize table with capacity = 1

for $i = 1$ to n :

if table is full:

newtable = create newtable
with size $2 \times \text{current size}$

copy elements from old
table to new table

table = newtable

Insert element i into table

Initialize charges = 0

Initialize credits = 0

for $i = 1$ to n :

charges $++$

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

credits $++m$

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

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

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

$$\begin{aligned}\text{Runtime per insertion} \\ &= O(1)\end{aligned}$$

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