HW - CH-17

Dhanush Pathipati
1002172455

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 to size 2, after 2 more insertions it doubles to size 4.

→ In general, after k doublings, the size is 2^k.

Psuedo code:

Initialize table with capacity = 1

for i = 1 to n;

if table is full:

new table = create newtable with size 2

copy elements from old table to new

table = new table.

insert element i into table.

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

Total cost $= O(n) * k$

$= O(n \log n)$

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 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 $O(m)$

→ 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 to n:

new table = create new table with size 2 * current size

copy elements from old table to new table

table = new table

insert element i into tabb.

initialize charges = 0

initialize credits = 0

for i = 1 to n

charges t = 2

if table doubles in size from m to 2 * m

Credits + = m

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

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

Amortized cost per insertion $= \text{Total} / n$

$= O(n) / n$

$= O(1)$

Runtime per insertion $= O(1)$

Total time $= O(n)$