Namra Patel
1002229720

DAA H.W-17

① Inserting n elements using

a) Aggregate Method.

→ The table doubles in size when it run out of space
→ So, if the original size is, after insertion it doubles to size 4 etc.
→ In general, after k doubling 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 cost element from old 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)

Amortise cost per insertion = O(log n)

Runtime per insertions is O(log n)

Total time in O(n) to log (n+1)

5) Accounting method.

* charge 2 units per each insertion

~> when the table double in side from m to 2m credit m units

~> The credit executes pay for copy cost at O(n).

~> Total credit is $m + 2m + nm + \ldots \frac{n}{2}m = O(n)$

Pseudo code:

~> initialise table with capacity = 1

for i = 0 to n;

if table is full:

newtable = create with sign zin current size

copy element from d table to new table.

table = newtable.

insert element i into table

initialise charges = 0

initialize credits = 0

for i = 1 to n;

charges += 2

if table doubled in size

from m to <m:

credits += m

total charges = 2 to n = O(n)

total credits = $m + 2m + \ldots + n/2\ m = O(n)$

Amortized cost per insertion:-

total / n. O(n)/n

= 1

Runtime per ingertion = O(1)

total time = O(n).