

Viraj Pathak

1002195291

HW-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 and 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^{th} current size

copy elements then from old table
to new table

table = new table

Insert element 1 into table

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

Total cost = $O(n)K$

$= O(n \log n)$

cost per insertion = $O(\log n)$

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

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

Date _____
Page _____

b) accounting methods:

charge 2 units for each insertion
When the table doubles in size from m to $2m$,
credit m units,

Total credit $\geq m + 2m + 4m + \dots$

$$\frac{1}{2} * m = O(n)$$

Pseudo Code :

Initialize table with capacity = 1

for $i = 1$ to n

if table is full:

new table = create new table

with size \times current size

copy element from old table to new table

table = new table

insert element i into table

Initialize charges = 0

Initialize 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 = $m + 2m + \dots$
 $= \frac{1}{2} * m = O(n)$

Cost per insertion = total / $n = O(n) / n$
 $= O(1)$

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