2242 - CSE - 5311 - 006

DSGN & ANLY ALGORITHMS

1002194693
A. M. N. Rajasekhar

chapter - 17

(1) Given a dynamic table that double in size when it needs more space. Find the amortized runtime for inserting n elements

(A) use the aggregate method

To insert 'n' elements using the aggregate method with can be done in 2 ways.

case 1 :-

If we donot take and need to allocate new memory then

$O(1)^1$ $O(1)^2$ $O(1)^3$ $O(1)^4$ $O(1)^5$ $O(1)^6$ $O(1)^7$ $O(1)^8$

so the sequence of n inserts

$$O(n) + O(2n) = O(n)$$

so, replace $O(1)$ in above example

$$O(1) + O(2n) = O(1)$$

thus the amortized runtime is $O(n)$ for inserting n elements is $O(1)$

for case 2 :-

If we allocate new memory

$i = 2^k + 1$, $k = 1, 2, 3 \ldots$,

to include the capacity & double the size of array

then we need to allocate new memory

For inserting the element n in new array

Runtime time $= 2^k + 1$ if $i = 2^{k+1}$ case 1

$= 1$ otherwise case 2.

(b) use the according method

using the accounting method. charge 2 units to each insertion.

when the table double in size from m to 2m units.

The credits exactly pay for the copy cost of O(m)

total credit is $m + 2m + 4m + \cdots$

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

Pseudo code :-

initialize table with capacity = 1

for i = 1 to n

if table in new table with size

2 * current size

insert element in into table

initalize charges = 0

initialize credits = 0

for i = 1 to n

charges += 2

credit += m

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

total credit $= m + 2m + \cdots n/2 + m = O(n)$

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

∴ runtime per insertion $= O(1)$

total time for inserting n elements is $O(n)$.