DAA

chapter - 17

Homework

(1.) Given a dynamic table that doubles in size when it needs more space. Find the amoritized 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.

For case 1:

If we don't take and need to allocate new memory then

1	2	3	4	5	6	7	8 ...
O(1)	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)

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 amoritized runtime is O(n) for inseeting n elements is O(1)

For case 2:

If we allocate new memory

$i = 2^k + 1$, $k = 1, 2, 3 ---$

to include the capacity & double the size of array

then we need to allocate new memory

For inserting the element n in the new array

$$\begin{cases} \text{Running time} = 2^k + 1 & \text{if } i = 2^{k+1}\text{, case 1} \\ = 1 & \text{otherwise, case 2} \end{cases}$$

(b) use the according method.

using the accounting method. change 2 units to 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 $m + 2m + 4m + - - -$

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

Pseudo code:

```
initialize table with capacity = 1
    for i = 1 to n:
  if table is new table with size
  2* current size
  insert element i into table
      initialize charges = 0
      initialize credits = 0

  for i = 1 to n
      charges + = 2
      credit + = m
  Total charges = $2 * n = O(n)$
  total credit = $m + 2m + -- n/2 + m = O(n)$
  cost per insertion = total/n = $O(n)/n = O(1)$
∴ runtime per ineseution = $O(1)$
Total time for inserting n elements is $O(n)$
```