Design Analysis and Algorithms

Homework - 17.

1 Aggregate Method

The aggregate method calculates the total cost of operations and divides by the number of operations to find the average cost per operation.

- Suppose we have dynamic array that starts with a capacity of 1.
- Each time we insert an element and array is full, array size doubles requiring copying all elements to the new array.

To analyze the cost.

Single Insertions: Inserting an element costs O(1) unless resizing is required.

Resizing: Each time resizing happens, all elements need to be copied to the new array, so cost of resizing after k insertions is $O(k)$.

When we insert n elements, total cost consists of both the simple $O(1)$ cost for each insertion and costs each resizing operation.

The total cost of inserting n elements is

$$\sum_{i=0}^{\log n} O(2^i) = O(n)$$

Thus, amortized cost per insertion is $O(1)$.

b) Accounting Method:

The accounting method es used to analyze the amortized cost of asequence of operations by charging some operations more than their actual cost and saving the excess as credits.

Pseudocode:

```
from i=1 to n
    if table es full
        newtable = Create newtable with size 2*
                                        current size
    then copy elements from old table tonewtable
        table = New table
        insert element i into table
            initate charge = 0
            for i ~~to~~ = 1 to n
                charges += 2
        if table double in size from m to 2m
            credits += m
```

Total charge = $2^{*} n = O(n)$

Total credit = $m + 2m + \text{-----} + n/2^{*} m = O(n)$

Amortized cost per insertion = Total/n

$= O(n/n)$

$= O(1)$

Run times pee insertion = $O(1)$

Total time = $O(n)$