Mathan Raj Arumugam Rajkumar
1002231625
Hands On - 11

a) Aggrigrate method:

Cost of insertion:
(i) each insertion typically costs $O(1)$.
(ii) If the array is full, resizing occurs at a cost of $O(n)$

Total cost:
(i) inserting 1st element, at a cost of 1, 2nd element insertion costs 2 because of the resizing of an array.
For inserting the 3rd element and 4th element costing 4 for copying the existing elements and cost 1 for insertion.

$\therefore$ the total cost of all resizing operations for n elements:

$$1+2+4+8+\cdots+2^{\lfloor \log n \rfloor}$$

which is a geometric series.

So,

$$1+2+4+8+\cdots+n/2 = n-1$$

Total cost for n insertion is,

$$T(n) = O(n).$$

Therefore, the amortized cost per insertion is,

$$\Rightarrow \frac{T(n)}{n} = \frac{O(n)}{n} = O(1).$$

b) Using the accounting method, assigning a charge of 3 units to each insertion,

1 unit covers the cost of regular insertion if there is only space.

2 units are saved used for future resizing operations.

When the array doubles in size, the cost of copying elements is fully covered by the credits saved from previous insertions.

Each element only gets copied a limited no. of times, so the amortized cost per insertion remains constant.

∴ The amortized runtime for inserting elements into a dynamic array that doubles in size is $O(1)$ per insertions, that leads to a total time of $O(n)$ for n insertions.