Hands-ON 12
Chapter-17

1. Dynamic Table Behaviour:-

* The table starts with an initial size, let's say m.
* When inserting an element into the table and it's full, it doubles its size.
* Doubling the size means creating a new table with double the size, copying all elements over, and then inserting the new element.
* Let's assume that inserting an element into the table of size m takes O(m) time.

(a) Aggregate method:

In the Aggregate method, we calculate the total time for a sequence of operations and then divide by the number of operations to find the average time per operation.

1. No. of operations(n) : n (insertions)
2. Total time : Let's consider each insertion:
   * The first insertion takes O(1) time.
   * The second insertion takes O(2) time.
   * The third insertion takes O(4) time.
   * And so on, until the table reaches size m', where m' is the smallest power of 2 larger than n.
     So, $m' = 2^{[\log_2(n)]}$.
   * Total time for all insertions would be: $1+2+4+\ldots+m'$.
   * This is a geometric series with the sum $2^{k+1}-1$, where $k = [\log_2(n)]$.
   * So, total time is $O(2^{k+1}-1)$.

3. Average time per operation: Divide the total time by n:
   * Average time per operation = $\frac{O(2^{k+1}-1)}{n}$.

(b) Accounting method:

In the accounting method, we assign a "credit" or "cost" to each operation, such that the total cost is an upper bound on the total actual cost.

1. Assigning Costs: Let's assign a cost to each insertion operation:
   * When we insert an element, we pay 2 units of cost: 1 unit for the insertion and 1 unit for potential future doubling.
   * This extra unit goes towards covering the future doubling cost.

2. Total cost: The total cost for all insertions would be 2n.
3. Average Cost per Operation: Divide the total cost by n:
   * Avg cost per operation = $\frac{2n}{n} = 2$.
   * This means each insertion operation amortizes to O(1) time.

Conclusion :- In both the methods, average time or cost per insertion is O(1).