

Hands-On 11

1] Gives a dynamic table (see section 17.4) that doubles in size when it needs more space. Find the amortized runtime for inserting n elements.

a) Use the aggregate method
In the aggregate method we consider the total cost across all the insertions and calculate the average (amortized) cost per insertion.

When inserting the i^{th} element, if a resize operation is not needed then the existing happens cost $O(1)$ as it involves copying the existing elements to the new table of size 2^K (K is the number of resizes performed.).

$$\begin{aligned}\text{Total cost} &= O(n)K \\ &= O(n \log n)\end{aligned}$$

$$\text{Cost per insertion} = O(\log n)$$

$$\text{Runtime per insertion} = O(\log n)$$

$$\text{Total time is } O(n) * \log(n+1).$$

b) Accounting Method

In the accounting Method, we assign each insertion a higher "amortized" cost to store "credits" that pay for future resizing costs.

Pseudo code:-

for $i \geq 1$ to n .

If table is full

new-table = create new table
with size then copy elements
from old table to new-table.

table = new-table

insert element i into table

initial charge = 0

for $i \geq 1$ to n .

charges += 2

If table doubled in size from
into $2m$.

credits += $2m$.

Total charge = $2^* n = O(n)$.

total credits = $m + 2m \dots n/2^* m$
 $= O(n)$.

Amortized cost per insertion

= Total / n .

= $O(n/n)$.

= $O(1)$.

runtime per insertion $O(1)$.

Total time $O(n)$.