

1. a. Aggregate method:

$$\frac{\text{total cost of } n \text{ ops}}{n} = \text{Avg cost per ops}$$

So: we insert w/o increasing space!

(Quicksort is $O(n \lg n)$ if unsorted)

BUT! we are not sorting.

So, should take $O(1)$ per insertion
total = $O(n)$

• increasing space: $O(c)$ where

c is size of memory you need
to allocate. let's say: $O(1)$ for now.

$$\text{total} = O(n)$$

$$\text{So: } \frac{O(n) + O(n)}{2} = O(n); \text{ and } T(n) = O(1)$$

b, ^{allocating} actual cost:

per insertion $O(1)$

increase space: $O(n)$

c, guess:

insertion: $O(n^2)$

increase space: $O(n)$

initialize table with size (n)

for $i=1$ to n ... $O(1)$

if $i \leq n$ $n = \text{table size}$

new table size = $2n$... $O(1)$

move each element to new array ... $O(n)$

Charge $t=2$

$$C_{\text{charge}} t = m + \dots + m = 0$$

Charge $t=2$

$$\text{So total charge} = 2n = O(n)$$

$$\text{total credits} = m + m + \dots + m = O(n)$$

$$(\text{arr}^M \text{ per credit}) = \frac{\text{total}}{n} = \frac{O(n)}{n} = O(1)$$

large number of credits

(total credits) = O(n)

But we are not done

total credits = O(n)

where (O(n) credits)

2 sets of credits

$$(\text{arr}^M = \text{total})$$

$$(\text{arr}^M) = \frac{(\text{arr}^M) + (\text{arr}^M)}{2}$$

total credits = O(n)

(O(n) credits)

(O(n) credits)

(O(n) credits)

(O(n) credits)

(O(n) credits)

(O(n) credits)