Amortized Cost

19 December 2021 20:43

Stack

1. push()

push (x)

$n \to$ Top

if $n == a.size$

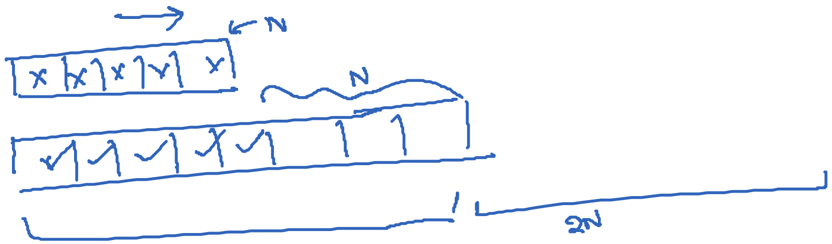
$a' = new\ arr[2 \times n]$

$a'[0 \dots n-1] = a[0 \dots n-1]$

$a = a'$

worst case $O(n)$

$O(1) \leftarrow a[n+1] = x;$ } $\to$ Usually this thing



op $\to$ operation

Amortized Analysis:

$T(op) \to$ real time

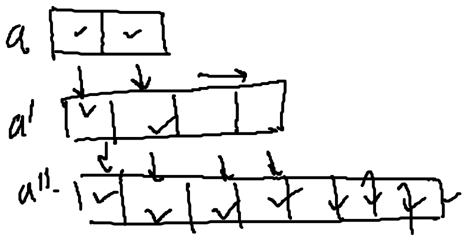
$\tilde{T}(op) \to$ Amort Time

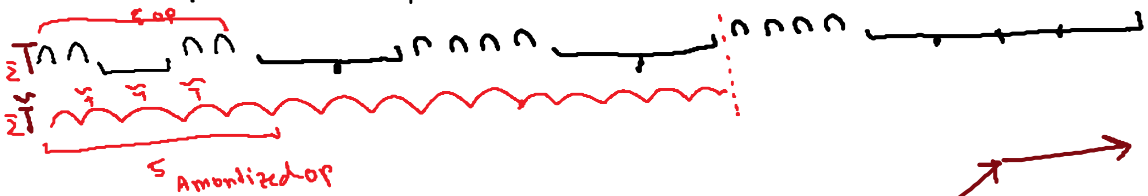
kind of average

$\underbrace{O_1\ O_2\ O_3\ O_4 \dots O_m}_{m\text{-operation}}$

$$\sum T(O_i) \leq \sum_{i=1}^{m} \tilde{T}(O_i)$$

upper bound





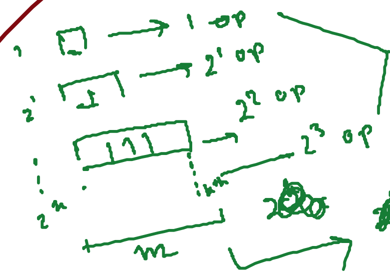
$\sum T(5 \text{ real op}) \leq \sum \tilde{T}(5 \text{ amortized})$

- Push Operation:

$\tilde{T}(push) = O(1) = c \Rightarrow \sum_{i=1}^{m} T(push) \leq c \cdot m$

Naive Proof:

$\sum T(O_i) = m +$



$\sum 2^i = 2^{k+1} - 1$

$m \geq 2^k$

$\therefore 2^{k+1} \leq 2m$

$\hookrightarrow \sum T(O) \leq 3m \to \tilde{T}(O_i) \leq c = 3$

push (x)

if $n == a.size$

$a' = new\ arr[2 \times n]$

$a'[0 \dots n-1] = a[0 \dots n-1]$

$a = a'$

$a[n+1] = x;$ } $O(1) \to$ This has to be done anyway

$\hookrightarrow m \cdot O(1)$

m elements are pushed

Potential Method :- (Universal)

$\phi()$ → state of current DS (How bad your DS is)

$o_1 \quad o_2 \quad o_3 \quad o_4 \quad o_5 \cdots o_m$

$\phi_0 \quad \phi_1 \quad \phi_2 \quad \phi_3 \quad \phi_4 \cdots \phi_m$

Say, $\phi_0 = 0$ and $\phi() \leq 0$ (Negative)

$\tilde{T} = T + \Delta\phi \; (\phi_{i+1} - \phi_i)$

$\sum \tilde{T} = \sum T + \sum \Delta\phi = \sum T + (\phi_m - \phi_0) = \sum T + \phi(m)$

Good DS → means $(\tilde{T} = T)$

How to find $\phi()$? → Lets look at the

To find a ϕ ... we ... su... he...

big ϕ_1 ↓ small ϕ_2

n n

x x x x

x x x x

n

→ Real Time is $T_1 = n$ for copying → Optimis... to re... real t...

So we want $\tilde{T}_1 \approx T_1$ → $\phi_2 - \phi_1 \approx -n$

$\therefore \Delta\phi < (-n)$

$\phi() \propto$ # elements in the right half $\propto -n/2$

i.e decreasing in later version

and, we know, $\phi_2 - \phi_1 = n$ → $\phi() = 2 \cdot x$ (# elements in the right half)

$T = n$ Time taken

$\Delta\phi = -n$ And, decrease potential by

$\hookrightarrow O(\tilde{T}) = 1$

Slightly Intuitive Technique :-

(Counting Method)

Aggregate Technique :-

$O(\sum \text{cheap} + \sum \text{Exp})$

$\tilde{T} =$

(Banking'

Total Time

#op

#op |
cost | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

2 4 1 1 1 8 1 1 1 1 1 1 1

$O(2n) \longrightarrow$ per op $O(1) \leftarrow$ Amort