→ Different design paradigm

→ What is amortized analysis?

→ Find how it works

→ Two examples. Stack & Binary counter
push, pop, Multipop → # of bit flips

→ Aggregate analysis
→ For a sequence of n operations takes $T(n)$ time
⇒ amortized cost per operation is $\frac{T(n)}{n}$.
diff types of operations, different costs.

→ Stack analysis → only push & pop $\Theta(n)$ → $\Theta(1)$ per op.

push, pop & multipop → $\Theta(n^2)$ → $\Theta(n)$ per op

# of pops ≤ # of pushes.
# of push ≤ n ⇒ # pops ≤ n
⇒ $O(n)$ total cost
⇒ $O(1)$ amortized cost

→ No probabilistic reasoning.

Incrementing a binary counter

Counter value		Total cost
0	0 0 0 0 0	0
1	1	1
	⋱	
16		31

k-bits. Counting the # of bit flips.

A single op could cost $\Theta(k)$ bits

→ sequence takes $\Theta(nk)$

Clever observation: A[i] flips $\lfloor n/2^i \rfloor$ times

$$\sum_{i=0}^{k-1} \left\lfloor \frac{n}{2^i} \right\rfloor \le n \cdot \sum_{i=0}^{\infty} \frac{1}{2^i} = 2n$$

→ sequence of n Increments $= O(n)$

→ amortized cost per op $= O(1)$.

→ Show how situation changes if Decrement is involved

Accounting method

→ Assign differing charges to different ops

→ Amount charged to an operation is its amortized cost

→ May be more or less than actual cost.

→ If am. cost > act. cost it is called credit.

Used to pay for other operations in debit.

- c_i = actual cost & $\hat{c}_i$ = amortized cost of i^{th} operation

$$\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i, \quad \forall n.$$

$$\sum_{i=1}^{n} \hat{c}_i - \sum_{i=1}^{n} c_i \geq 0 \quad \text{at all times.}$$

Stack →	Actual	Amortized
Push	1	2
Pop	1	0
multipop	min (k, s)	0

eg. to see $O(n)$ total ⇒ $O(1)$ per op

Counter $2 to set a bit to 1; $ to reset to 0.

At most one bit is set to 1, in an increment operation

Potential Method

→ Assigning a potential to the data structure instead of specific objects.

Initial data structure D_0.

n operations. c_i = actual cost. D_i, i^{th} data structure.

$\phi : D_i \to \mathbb{R}$.

Amortized cost $\hat{c}_i = c_i + \phi(D_i) - \phi(D_{i-1})$

$$\sum_{i=1}^{n} \hat{c}_i = \sum_{i=1}^{n} c_i + \phi(D_n) - \phi(D_0)$$

We must have $\phi(D_i) \geq \phi(D_0)$ $\forall i$

Usually $\phi(D_0) = 0$ & $\phi(D_i) \geq 0$ $\forall i$.

→ More than one way to choose potential.

Stack operations

→ ϕ = # of objects in stack

$\phi(D_0) = 0$

$\phi(D_i) \geq 0$

If i^{th} operation on a stack of s objects is Push

$$\phi(D_i) - \phi(D_{i-1}) = (s+1) - s$$
$$= 1$$

⇒ Amortized cost $\hat{c}_i = 1 + 1 = 2$

If the i^{th} operation is Multipop(S, k)
$k' = \min(k, s)$ to be popped

$$\phi(D_i) - \phi(D_{i-1}) = -k'$$
$$\hat{c}_i = c_i + \phi(D_i) - \phi(D_{i-1})$$
$$= k' - k'$$
$$= 0$$

⇒ Total amortized cost of a sequence $O(n)$

→ $O(1)$ for a single operation