Amortized Analysis

2017년 11월 6일 월요일

오후 1:00

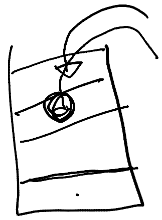
⇒ averge cost of a operation over worst-case seq. operation

1. Aggregate method:

Show that a seq of n op. takes $T(n)$ time in total. The avg. cost per op. is $T(n)/n$.

$T(n)$ = worst case analysis in seq. of n op.

ex 1) Stack with pop, push, Multi-pop.



~~Fak~~ LIFO

Multi-pop (S, K)
  while ($S \neq 0$ and $K \neq 0$)
    pop(S); K--;
  endwhile

actual cost of Multi-pop.

expense min (|S|, K)

* aggregate analysis

possible # of push/pop/multipop
in term of n # operations.

For any value n, any seq. of
n push/pop/multi-pop op. takes
$\Theta(n)$ time

$\therefore\ O(n)/n = O(1)$

ex2) Incrementing on binary counter.

0000
0001
0010
$2^3\ 2^2\ 2^1\ 2^0$
8 4 2 0

k-bit array A[0.... k-1] of bits where
the lowest order bit is A[0] and
length (A) = k as counter.

A counter hold a value x s.t

$$x = \sum_{i=0}^{k-1} A[i] \cdot 2^i$$

Initalize x = 0

Increment (A)

j = 0
while j < lenth (A) and A[j] = 1 do

A[j] ← 0.
j ← j + 1.

} Lot 0

~~while~~ endwhile

if j < length(A) then

$A[j] \leftarrow 1$) cost 1

endif

end Increment

Regular analysis

One increment is $\Theta(k)$ in the worst case

n Increments $O(nk)$

* better analysis

→ $A[i]$ flips $\lfloor n/2^i \rfloor$ times in seq of n Increments

counter	$\overset{2^3}{A[3]}$	$\overset{2^2}{A[2]}$	$\overset{2^1}{A[1]}$	$\overset{2^0}{A[0]}$	
0	0	0	0	0	+1
1	0	0	0	1	3)+2
2	0	0	1	0	4)+1
3	0	0	1	1	7)+3
4	0	1	0	0	8)+1
5	0	1	0	1	10)+2
6	0	1	1	0	

$\lg 6 \approx$

A[0] A[1] A[2] A[3]

6 | 0 1 1 0 1 0

7 | $n/2$ ③ ① 3 6 $\lfloor n/2^9 \rfloor$

$\lfloor n/2^2 \rfloor$ $\lfloor n/2^i \rfloor$

total # of flip in seq. op. n

$$= \sum_{i=0}^{\lfloor \lg n \rfloor} \lfloor n/2^i \rfloor$$

$$< n \cdot \sum_{i=0}^{\infty} 1/2^i$$

$$= 2n$$

$\therefore$ avg cost $O(n)/n = O(1)$

2. Accounting Method.

⇒ assign different "charges" to different operations

⇒ amortize cost : more or less than actual cost

- If cost is more than actual cost assign the surplus cost to specific object in the DS. as "credit" ≥ 0

ex1) stack

stack	idea 1 charge	idea 2 DS credit	charge
push	0		2
pop	1		0
multipop	k		0

cost: 0, 1, K

→ 10 push 1 pop multipop(k=9)

credit: ~~10~~ ~~9~~ 0

cost = ~~10~~ 11 + 9 = 20

ex2) Inc Binary counter

	Charge
① 0 → 1	2 (1+1)
② 1 → 0	0

3. Potential Method

perform n opr. from DS D.

for $i = 1 \dots n$, let c_i be the actual cost of i th operation and

D_i be the DS resulting from ith op.

A pontential $\Phi(D_i)$ is the potential value of D_i

⇒ amortized cost $\hat{c}_i$ of ith op.

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

⇒ Total amortize cost

$$\sum_{i=1}^{n} \hat{c}_i = \sum_{i=1}^{n} \left(c_i + \Phi(D_i) - \Phi(D_{i-1}) \right)$$

$= 0$

$$\boxed{\sum_{i=1}^{n} c_i}$$

ex) stack

potential ≡ # of elements in the stack

push : $c_i = 1$, $\overset{k+1}{\Phi(D_i)} - \overset{k}{\Phi(D_{i-1})} = 1$

push : $c_i = 1$, $\underbrace{\Phi(D_i) - \Phi(D_{i-1})}$ = 1

$\hat{c}_i = 1 + 1 = 2$

pop : $c_i = 1$ $\quad \underbrace{\Phi(D_i)}_{k} - \Phi(D_{i-1}) = -1$

$\hat{c}_i = 0$

multipop : $c_i = \underbrace{\min(|S|, a)}$ $= a$

$(S, \underline{a})$

$\overline{\underline{\Phi}}(D_i) - \underline{\Phi}(D_{i-1}) = -a$

$\qquad k-a \qquad\qquad k$

$\hat{c}_i = a - a = 0$

ex2) IBC

potential : # of 1's in the current binary representation

b_i

→ Suppose that ith Increment op will $\underbrace{\text{reset}}_{1 \to 0}$ t_i bit.

Cost of Increment

$c_i = t_i + 1$ $\qquad\qquad b_i = b_{i-1} - \underline{t_i} + 1$

$\underbrace{\Phi(D_i)} - \underbrace{\Phi(D_{i-1})}$

$$\Phi(D_i) - \Phi(D_{i-1})$$

$$\le (\cancel{b_{i-1}} - t_i + 1) - (\cancel{b_{i-1}})$$

$$= 1 - t_i$$

$$\hat{c}_i = \cancel{t_i} + 1 + 1 - \cancel{t_i}$$

$$= 2.$$