Ch. 17 Amortized Analysis

17.1 Aggregate Analysis

Let S be a stack with operations

Push(S, x) (push x onto stack S)

Pop(S) (pop top object off stack S)

Push and Pop have $O(1)$ run-time.

Suppose we also have the operation:

Multipop(S, k)

which pops the top k objects off the stack,

if S has fewer than k objects it just pops the objects in S

Multipop (S, k)
while not Stack-empty(S) and k > 0
    Pop(S)
    k = k-1

Note that the cost of Multipop(S, k) is $\min(s, k)$ where s is the number of objects on S.

and the worst-case running-time of Multipop is $O(\text{max-size of } S)$.

We want to analyse the running-time of a sequence of n operations of Push, Pop, Multipop on a stack S.

Push is $O(1)$
Pop is $O(1)$
Multipop is $O(n)$ (since there cannot be more than n objects on S)

Thus the worst-case operation is $O(n)$
and if we do n operations
this will be $O(n^2)$

This analysis is not very good since it does not take into account the fact that we cannot Multipop $O(n)$ objects without Pushing objects onto the stack.

A more accurate analysis is obtained using aggregate analysis in which the entire sequence is considered.

Starting from an empty stack, the number of pops we can do, including those in a multipop, is at most the number of pushes we can do, which is $O(n)$

Thus, the run-time of n operations is $O(2n)$, which is $O(n)$.

The <u>average</u> run-time of n operations

is $\dfrac{\Theta(n)}{n} = \Theta(1)$

The <u>amortized</u> cost is $\Theta(1)$ per operation

Incrementing a Binary Counter.

Consider a Boolean array A :

n increments

k bits	cost	Running cost.
0 0 0 0 0 0 0 0		
0 0 0 0 0 0 0 1	1	1
0 0 0 0 0 0 1 0	2	3
0 0 0 0 0 0 1 1	1	4
0 0 0 0 0 1 0 0	3	7
0 0 0 0 0 1 0 1	1	8
0 0 0 0 0 1 1 0	2	10
0 0 0 0 0 1 1 1	1	11
0 0 0 0 1 0 0 0	4	15

what is the run-time of n increment operations?

the worst-case single increment does k bit-flips, so for n the run-time is $O(nk)$.

Using aggregate analysis we can count the cost of n increments and divide by n to get amortized cost per increment.

Total number of bit-flips in sequence is

$$n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \dots + \frac{n}{2^{k-1}}$$

$$n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \cdots + \frac{n}{2^{k-1}}$$

$$= n \sum_{i=0}^{k-1} \frac{1}{2^i}$$

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

$$= 2n$$

Thus the average cost (amortized cost)

per increment is $\frac{O(2n)}{n} = \frac{O(n)}{n} = O(1)$