Chapter 17

17.1-1

No. A sequence like

		Cost
n operations	Multipush(S, n)	n
	Multipop(S, n)	n
	Multipush(S, n)	n
	Multipop(S, n)	n
	⋮	⋮
	Multipop(S, n)	n

The cost of the sequence of n operations is n^2, so the amortized cost is $\Theta(\frac{n^2}{n}) = \Theta(n)$

17.1-2

Once the counter is at:

$$\overbrace{011\cdots11}^{k \text{ bits}}$$

incr.	$100\cdots00$	cost = k
decr.	$011\cdots11$	cost = k
incr.	$100\cdots00$	cost = k
⋮		⋮

n operations, each with cost k, gives $\Theta(nk)$.

17.2-1

Note that only Push, Pop are allowed & then COPY which copies all items on the stack to backup drive.

Use amortized cost

$\hat{c}_i = 2$ if c_i is PUSH & if c_i is POP.

$\hat{c}_i = 0$ if c_i is COPY.

The amortized cost includes the actual cost of 1 for PUSH OR POP & the remaining 1 is kept aside to use for the copy operation. Then there is always enough credit to copy stack items.

Thus any n operations cost $O(2n)$
$= O(n)$

17.3-1

Define the function Φ' by

$$\Phi'(D_i) = \Phi(D_i) - \Phi(D_0).$$

Then $\Phi'(D_0) = 0$

& $\Phi'(D_i) \geq 0$ for all i,

so Φ' is a potential function.

The amortized costs associated with Φ' are:

$$\begin{aligned}
\hat{c}'_i &= c_i + \Phi'(D_i) - \Phi'(D_{i-1}) \\
&= c_i + (\Phi(D_i) - \Phi(D_0)) - (\Phi(D_{i-1}) - \Phi(D_0)) \\
&= c_i + \Phi(D_i) - \Phi(D_{i-1}) \\
&= \hat{c}_i
\end{aligned}$$

So the amortized costs for Φ' and Φ are the same.

17.3-4

Since there are s_0 elements on the stack at the start, and no credits to begin with, it costs 1 for each element popped off of the original s_0 element.

In the worst case, all s_0 elements are popped off using a single MULTIPOP, which has a cost of s_0.

Then the remaining $n-1$ operations are as before, with 2 for PUSH, 0 for POP and 0 for multipop.

Thus the cost of n operations is

$$O(s_0 + 2(n-1)).$$

which is $O(n)$

If the initial elements are popped off one at a time, then s_0 pops are needed each with cost 1. Then n operations costs

$$O(s_0 + 2(n-s_0))$$

which is also just $O(n)$

Note that the number of elements s_n on the stack at the end of n operations is irrelevant.