Example Question 1: Prove that the INCREMENT operation runs in $O(1)$ amortized time using (a) the accounting method and (b) the potential method.

1. In a $k$-bit binary counter, the INCREMENT operation flips the lowest order bit, and if that bit was a 1, then it flips the next lowest order bit, and so on until it either runs out of bits or it finds a 0, which it flips to a 1. The real cost of INCREMENT depends on the number of 1s that we flip before encountering a 0. In the worst case, all $k$ bits are 1 and we must flip all of them, yielding a worst-case complexity of $O(k)$. Below, I show how this operation can be done in $O(1)$ amortized time.

(a) Using the accounting method, we will charge each INCREMENT operation two coins. The first coin will pay for changing a 0 to a 1, which from our description above happens at most once in each increment. (The only time it does not happen is when all $k$ bits are already set to 1.) The second coin is placed on the new 1 that was just changed from a 0 (assuming there is one). This second coin pays for flipping that 1 back into a 0 in a future INCREMENT operation. Therefore, when we perform an INCREMENT, each of the flips from a 1 to a 0 has been paid for by the previous INCREMENT that made it into a 1. Therefore, the amortized cost of INCREMENT is $O(1)$.

(b) Using the potential method, we will define our potential $\Phi_i$ to be the number of bits that are set to 1 in the counter after a particular sequence of $i$ operations has been performed. Let $l$ be the number of 1s that are changed to 0s during the $i$th INCREMENT. The real cost of flipping $l$ 1s into 0s and one 0 into a 1 is $l + 1$. The change in the potential function is 1 (for the 0 to 1 flip) plus $-l$ (for the $l$ 1 to 0 flips), so the total amortized cost is as follows:

$$
\hat{c}_i = c_i + \Phi_i - \Phi_{i-1} = (l + 1) + (1 - l) = 2
$$

Since the potential is initially zero and is never negative, $2n$ is an upper bound on the cost of $n$ INCREMENTs and the amortized cost of INCREMENT is $O(1)$. 

1