Задача 3

Реальное время работы такого алгоритма равно $O(k)$. Например рассмотрим следующий шаг алгоритма: все биты счётчика равны единице, тогда в цикле выполнится $\Theta(k)$ действий.

Докажем, что учётное время работы равно $O(1)$. Обозначим за z_i число битов счётчика, равных единице после i-ой операции.

Введём потенциалы: $f_i = z_i \cdot c, \ c > 0$. Изначально все биты счётчика нулевые, поэтому $f_0 = 0$. Также по нашему определению $f_i \geqslant 0$.

Пусть t_i — реальное время выполнения i-ой операции, а a_i — учётное время.

$$a_i = t_i + f_i - f_{i-1}$$

Докажем, что $a_i = O(1)$:
Цикл $while\ (i \ < \ A.length() \ \&\& \ A[i] \ == \ 1)$ будет итерироваться пока $A[i]$ не станет равным нулю. Пусть k — индекс младшего нулевого бита до i-ой операции. Тогда $t_i = \Theta(k)$, $z_{i-1} - k = z_i - 1 \implies f_i - f_{i-1} = -k \cdot c + c$.

Значит $a_i = \Theta(k) - k \cdot c + c$. Мы всегда можем подобрать c чтобы перекрыть константу в Θ, поэтому $a_i = \mathrm{c} = O(1)$, то есть амортизационное время работы алгоритма равно $O(1)$.

Пусть теперь есть ещё операция **Decrement**. Рассмотрим следующую последовательность из $2n$ операций (изначально все биты счётчика нулевые): каждая операция с нечётным индексом (нумерация с 1) - это Decrement, с чётным - это Increment. Тогда реальное время выполнения каждой операции будет $\Theta(k)$, значит и учётное время работы будет тоже $\Theta(k)$.