Problem 6

Algorithm countZerosAndOnes (A, start, end)	*Count of operations*
Input: sorted array A of zeros and ones, starting index, ending index	
Output: count of zeros, count of ones	
if (start ≥ end) then	*1*
ones = A.length – start – 1	*3*
zeros = A.length – ones	*2*
return zeros, ones	*2*
mid = (start + end) / 2	*3*
if (A[mid] = 1) then	*2*
return countZerosAndOnes (A, start, mid)	*2 + T(n/2)*
else	
return countZerosAndOnes (A, mid +1 , end)	*3 + T(n/2)*

$$T(n) = \begin{cases} 8, & n = 1 \\ T\left(\frac{n}{2}\right) + 16, & n > 1 \end{cases}$$

According to the master formula:

$a = 1, b = 2, c = 16, k = 0$
$\therefore a = 1 = b^k = 2^0 = 1$
$\therefore T(n)\ is\ \Theta\left(n^k \log n\right) \rightarrow T(n)\ is\ \Theta(\log n)$

Since $\lim_{n\to\infty} \frac{\log n}{n} = \lim_{n\to\infty} \frac{1}{n} \log e = 0$
$\therefore T(n)\ is\ o(n)$.