We define T(n) as the time complexity of the merge sort
We define D(n) as the divide algorithm's time complexity
We define C(n) as the combine algorithm's time complexity

Thus

$$T(n) = \begin{cases} \Theta(n) & if\ n \leq\ c, \\ aT(n/2) + D(n) + C(n) & otherwise. \end{cases}$$

Beceause we are just computing length(a) :

$$D(n) =\ \Theta(1)$$

Because we are merging at max an array of n elements :

$$C(n) =\ \Theta(n) = cn$$

$$T(n) =\ 2T(n/2) + cn$$

if n > 1

$$T(n/2) =\ 2T(n/4) + cn/2 \Rightarrow T(n) = 4T(n/4) + 2 * cn$$

$$having\ T(n/2^{log(n)}) = c\ and \sum T(n/2^i) = cn + 2^{i+1}T(n/2^{i+1})$$

$$\Rightarrow T(n) = cn(log(n) - 1) + 2 * cn$$

$$\Rightarrow T(n) = cnlog(n) + cn$$