Cálculo de la eficiencia del algoritmo basado en divide y vencerás usado para el ejercicio 2.

Vamos a considerar que evaluamos un vector de n elementos, siendo n potencia de 2 (2^k).

En el peor caso:

$T(n) \begin{cases} c1 & si\, n=1 \\ 2T(n/2)+c_2 & si\, n>1, n=0 \end{cases}$

Para saber la eficiencia, vamos a usar expansión:

$T(n) = 2T(n/2) + c_2$

$T(n/2) = 2T(n/4) + c_2$

Es decir: $T(n) = 4T(n/4) + 2c_2$ o $T(n) = 8T(n/8) + 3c_2$

En general: $T(n) = 2^iT(n/2^i) + ic_2$, siendo i el número de llamadas recursivas.

Cuando $i = k$, quiere decir que no habrá más llamadas recursivas, es decir, en la parte derecha hay $T(1)$.

La fórmula quedaría: $T(n) = 2^kT(1) + kc_2$

Como $2^k = n$, $k = log_2(n)$

La fórmula finalmente quedaría: $T(n) = nT(1) + c_2nlog_2(n)$

Podemos tomar $T(1) = c_1$ Por lo que la fórmula quedaría $T(n) = c_1n + c_2nlog_2(n)$, por lo que la eficiencia sería $O(nlog(n))$.