

Cálculo de la eficiencia del algoritmo basado en fuerza bruta usado para el ejercicio 3.

Vamos a considerar que evaluamos un vector de  $n$  elementos, siendo  $n$  potencia de 2 ( $2^k$ ). En el peor caso:

$$T(n) \begin{cases} c_1 & \text{si } n=1 \\ T(n/2)+c_2 & \text{si } n>1, n=2^k \end{cases}$$

Para saber la eficiencia, vamos a usar expansión

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

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

Es decir:

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

o

$$T(n) = T(n/8) + 3c_2$$

En general:

$$T(n) = T(n/2^i) + ic_2, \text{ siendo } i \text{ el número de llamadas recursivas.}$$

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

La fórmula quedaría:

$$T(n) = T(1) + kc_2$$

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

La fórmula quedaría:

$$T(n) = T(1) + c_2 \log_2(n)$$

Podemos tomar  $T(1) = c_1$

Por lo que la fórmula quedaría  $T(n) = c_1 + c_2 \log_2(n)$ , por lo que la eficiencia sería  $O(\log(n))$ .