

Desarrollo detallado de la recurrencia de Búsqueda Binaria

Sea $T(n)$ el tiempo de ejecución en el peor caso para buscar en un arreglo de tamaño n . La recurrencia es:

$$T(n) = T\left(\frac{n}{2}\right) + c, \quad T(1) = d.$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + c \\ &= \left[T\left(\frac{n}{2^2}\right) + c\right] + c = T\left(\frac{n}{4}\right) + 2c \\ &= \left[T\left(\frac{n}{2^3}\right) + c\right] + 2c = T\left(\frac{n}{8}\right) + 3c \\ &= \left[T\left(\frac{n}{2^4}\right) + c\right] + 3c = T\left(\frac{n}{16}\right) + 4c \\ &\vdots \\ &= T\left(\frac{n}{2^k}\right) + k c. \end{aligned}$$

Para que la recursión llegue a la base, imponemos

$$\frac{n}{2^k} = 1 \quad \implies \quad 2^k = n \quad \implies \quad k = \log_2 n.$$

Sustituyendo:

$$T(n) = T(1) + c \log_2 n = d + c \log_2 n.$$

Cotas asintóticas:

$$T(n) = c \log_2 n + d \quad \implies \quad T(n) \in \Theta(\log n), \quad T(n) \in O(\log n), \quad T(n) \in \Omega(\log n).$$