

Análisis de Complejidad de Búsqueda Binaria Recursiva

Descripción del algoritmo

La búsqueda binaria es un algoritmo eficiente para encontrar un elemento dentro de un arreglo **ordenado**. El algoritmo compara el valor buscado con el elemento central del arreglo:

- Si el valor es igual al del centro, se retorna el índice.
- Si el valor es menor, se busca recursivamente en la mitad izquierda.
- Si el valor es mayor, se busca en la mitad derecha.

En cada paso, el tamaño del problema se reduce a la mitad, lo que da lugar a una complejidad logarítmica.

Planteamiento de la función de recurrencia

Sea $T(n)$ el tiempo que tarda la búsqueda binaria sobre un arreglo de tamaño n .

- Cada llamada realiza una comparación y calcula el índice medio: esto toma tiempo constante $O(1)$.
- Luego realiza una única llamada recursiva sobre un subarreglo de tamaño $n/2$.

La función de recurrencia queda como:

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

Donde c es una constante.

Resolución de la recurrencia

Aplicamos el método de expansión:

$$\begin{aligned}
T(n) &= T(n/2) + c \\
&= T(n/4) + c + c = T(n/4) + 2c \\
&= T(n/8) + 3c \\
&\vdots \\
&= T(n/2^k) + kc
\end{aligned}$$

La recursión termina cuando $n/2^k = 1$, es decir:

$$2^k = n \Rightarrow k = \log_2 n$$

Sustituyendo:

$$T(n) = T(1) + c \cdot \log_2 n$$

Dado que $T(1)$ es constante, concluimos que:

$$T(n) \in O(\log n)$$

Análisis por casos

- **Mejor caso:** el elemento buscado se encuentra justo en el centro en la primera comparación.

$$T(n) \in O(1)$$

- **Peor caso:** el elemento no está en el arreglo o se encuentra en los extremos. Se realiza la máxima cantidad de divisiones hasta que el subproblema tiene tamaño 1.

$$T(n) \in O(\log n)$$

- **Caso promedio:** se necesitan varias divisiones hasta encontrar el elemento. Como el número de divisiones es logarítmico, se mantiene:

$$T(n) \in O(\log n)$$

Conclusión

La búsqueda binaria recursiva es muy eficiente sobre arreglos ordenados, con complejidad temporal logarítmica en el peor y caso promedio. Se destaca por su simplicidad y velocidad, pero requiere que los datos estén previamente ordenados.