**Demostración de la complejidad logarítmica del algoritmo de Búsqueda Binaria**

El algoritmo de Búsqueda Binaria consiste en encontrar un determinado número dentro de un arreglo ordenado, para lo cual se va dividiendo el arreglo en mitad y analizando si el valor buscado está en la parte inferior o superior del arreglo, de esa forma en cada iteración hasta encontrar el elemento buscado.

La complejidad de este algoritmo e logarítmica, lo cual vamos a demostrar partiendo de un ejemplo de las iteraciones que se pueden dar teniendo en cuenta que tenemos un arreglo de “n” elementos:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | …… | n |

Iteración 0:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | …… | n/2 |

Iteración 1:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | …… | n/4 |

Iteración 2:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | …… | n/8 |

Iteración 3:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | …… | n/16 |

Iteración 4:

Poniéndonos en el peor de los casos y que el valor buscado es el “1”. Tal como se muestra en las iteraciones, tenemos que dividir el valor de “n” tantas veces sea necesario hasta llegar a encontrar el valor de “1”, esto teniendo en cuenta que el arreglo utilizado debe estar ordenado.

El valor de las mitades va cambiando en cada iteración, de tal forma que podemos inducir lo siguiente:

Podemos observar que, por cada iteración, el factor constante de mitad es el siguiente:

El factor d mitad en algún momento debe llegar a ser igual al valor buscado en este caso es el “1”, por lo que tenemos lo siguiente.

Log2n = x

Con ello se demuestra que el ordenamiento binario tiene una complejidad logarítmica, según la notación Big-O se tiene lo siguiente:

**Notación Big-O** = O(log(n))

Generalizando la forma en la que está limitado el valor de “N” tenemos:

Podemos validar en forma general que el valor de k(0, 1, 2, 3,…) será menor que el logaritmo de N, lo cual permite validar que la complejidad del algoritmo de ordenamiento binario es logarítmico.