**-¿Por qué el algoritmo de Búsqueda Binaria tiene complejidad logarítmica?**

El Algoritmo de búsqueda binaria es un algoritmo de búsqueda en vectores ordenados que permite disminuir la complejidad de la búsqueda en dichos vectores. Consiste en mantener unos índices que marcan los límites superior e inferior de la parte del vector que nos queda por analizar. Para calcular la complejidad de este algoritmo, el número de elementos por analizar es n. Tras la primera división, el número será como mucho n/2, luego n/4, y así sucesivamente. Tras una división i, el número de elementos será, como mucho:

*\[\frac{n}{2^i}\]*

El peor caso se da cuando el elemento buscado no está en el vector (cuando el número de elementos a analizar es menor a 1). Por ejemplo, el número de llamadas a realizar es el número m tal que:

*[\frac{n}{2^m} < 1\].*

Transformando ésta fórmula a un logaritmo base 2, resulta:

*\[n < 2^m\],*

Y que:

*\[\log n < m\].*

Por ésto es que el algoritmo de Búsqueda Binaria tiene complejidad logarítmica.