

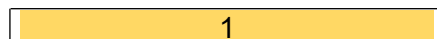
Complejidad del Algoritmo de Búsqueda Binaria

El algoritmo de la búsqueda binaria consiste en los siguientes pasos:

1. Dividir el rango o lista de elementos en un punto medio.
2. Se realiza una comparación entre el valor buscado y el valor ubicado en el punto medio.
3. De ser diferentes se procede a buscar en la zona donde existen más posibilidades que se encuentre dicho dato (Derecha o izquierda).
4. Repetimos nuevamente el paso 1 hasta encontrar o no el valor deseado.

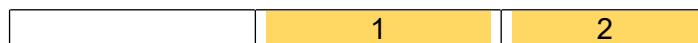
Entonces aplicando lo indicado se tiene los siguientes casos para un conjunto de N elementos donde para ciertos valores de n se tiene lo siguiente:

- $N = 1$



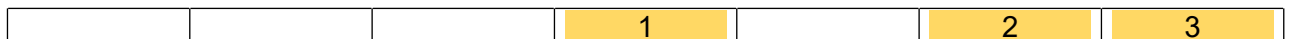
Se realizan 1 Comparación como máximo.

- $N = 3$



Se realizan 2 comparaciones como máximo.

- $N = 7$



Se realizarán 3 comparaciones como máximo.

- $N = 15$



Se realizarán 4 comparaciones como máximo.

De lo encontrado se puede formar la siguiente tabla:

| n | Comparaciones (m) |
|----|-------------------|
| 1 | 1 |
| 3 | 2 |
| 7 | 3 |
| 15 | 4 |

De la tabla podemos deducir que para n se realizaran $2^m - 1$ donde m es la cantidad de comparaciones a realizar, se procede a despejar m .

$$n = 2^m - 1$$

$$n + 1 = 2^m$$

$$\log_2 (n + 1) = \log_2 2^m$$

Por lo tanto se tiene que: **$m = \log_2 (n + 1)$** donde **$m \geq 1$** y m representa la cantidad de comparaciones a realizar.

Asumiendo que siempre se podrá obtener un número n que sea aproximadamente $(2^m - 1)$, para un N grande se tiene que la complejidad será:

$$\mathbf{O(n) = \log_2 (n + 1)}$$

Por lo tanto la complejidad de dicho algoritmo es logarítmica.

$$\mathbf{Complejidad = O(\log_2 n)}$$