**Complejidad del Algoritmo de Busqueda Binaria**

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

1. Dividir el rango o lista de elementos en un punto medio.
2. Se realiza una comparacion entre el valor buscado y el valor ubicado en el punto medio.
3. De ser diferentes se procede buscar en la zona donde existen mas 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

|  |
| --- |
| 1 |

Se realizan 1 Comparacion como maximo.

* N = 3

|  |  |  |
| --- | --- | --- |
|  | 1 | 2 |

Se realizan 2 comparaciones como maximo.

* N = 7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 1 |  | 2 | 3 |

Se realizaran 3 comparaciones como maximo.

* N = 15

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | 1 |  |  |  | 2 |  | 3 | 4 |

Se realizaran 4 comparaciones como maximo.

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 - 1 donde m es la cantidad de comparaciones a realizar, se procede a despejar m.

n = - 1

n + 1 =

=

Por lo tanto se tiene que: **m =**  donde **m > = 1** y m representa la cantida de comparaciones a realizar.

Asumiendo que siempre se podra obtener un numero n que sea aprimadamente ( -1), para un N grande se tiene que la compejidad sera:

**O (n) =**

Por lo tanto la complejidad de dicho algorito es logaritmica.

**Complejidad = O(n)**