Iteration 1

Length of the array = n

Iteration 2

The array is divided in two

Length of the array =

Iteration 3

*The chosen half is divided in two*

Length of the array = = =

Iteration 4

*The chosen half is divided in two*

Length of the array = = =

\*We have found a pattern, now we can know the length of the array after the k iteration\*

After Iteration k

Length of the array =

\*After k divisions the length of the array is 1. Because we divide, divide and divide the array until we find (or not) the number\*

= 1

=

=

To get the time complexity (Big O) we need the value of k. So, we apply the binary logarithm on both sides to have the k alone on one side later.

=

We can apply the power rule property of logarithms, that way, we can extract the k from the logarithm.

|  |
| --- |
| \*Power rule\* |

=

Now, we can apply the identity rule to get rid of the

|  |
| --- |
| \*Identity rule\* |

=

The complexity time of Binary Search is

O()