Como sabemos que una búsqueda lineal en un arreglo o vector de n elementos toma hasta n intentos encontrar el elemento. Por lo cual utilizar una búsqueda binaria (recursiva) hace menos intentos que una búsqueda lineal, la diferencia se hace más notable cuando el arreglo o vector aumenta de tamaño.

El propósito de esta búsqueda en reducir razonablemente el número de intentos, ejemplo si tenemos un vector de 40 elementos, en los intentos incorrectos que pudiera tener en el peor caso sería a lo más 20 los intentos, se reducen a la mitad.

Cada vez que duplicamos el tamaño del arreglo, necesitamos a lo más un intento más. Supón que necesitamos a lo más m intentos para un arreglo de longitud n. Entonces, para un arreglo de longitud 2n, el primer intento corta la porción razonable del arreglo a un tamaño n, y a lo más en m intentos terminamos, dándonos un total de a lo más m+1 intentos.

En el caso general de un arreglo de longitud n. Podemos expresar el número de intentos, en el peor caso, como "el número de veces que podemos repetidamente reducir a la mitad, empezando en n, hasta que obtengamos el valor 1, más uno". Pero eso es inconveniente para escribir. Afortunadamente, hay una función matemática que significa lo mismo que el número de veces que repetidamente reducimos a la mitad, empezando en n, hasta que obtenemos el valor de 1: el logaritmo base 2 de n. Lo escribimos como lgn.