

## Casos da Busca Binária

Como você deve saber, a complexidade temporal da busca binária, no pior caso, é  $O(\log n)$ . Assim, se uma lista contiver  $n$  elementos, então no pior caso serão acessados aproximadamente  $\log_2(n)$  elementos.

Neste exercício você deverá elaborar um programa que verifica isso. O código inicial contém uma implementação do algoritmo de busca binária em Python. Ele recebe como entrada uma lista e uma chave. Se houver um elemento na lista com a chave desejada, então o índice desse elemento é retornado. Caso contrário o algoritmo retorna o valor **-1**.

Modifique esse algoritmo para retornar o número de elementos da lista que foram verificados durante a busca. Depois escreva um programa principal que lê uma lista e verifica quantos elementos são verificados no pior caso.

## Entrada e Saída

A entrada será uma lista contendo  $N$  elementos inteiros em ordem crescente. Imprima o número de elementos que são verificados, no pior caso, para realizar busca binária na lista de entrada.

## Exemplos de Entrada e Saída

Entrada	[2, 3, 5, 7]
Saída	<sup>3</sup>
Entrada	[0, 1, 2, 3, 5, 8, 13, 20, 25, 32, 36, 41, 42, 57, 60, 68]
Saída	<sup>5</sup>