Busca Binária Recursiva

> *Uma busca binária é um método de pesquisa por elementos em vetores com acesso aleatório (isto é, é possível acessar elementos diretamente pela sua posição) e ela exige que o vetor já esteja ordenado. A busca realiza sucessivas divisões do vetor (identificando seu ponto central) para descartar a metade dos elementos em cada etapa. Se o elemento procurado estiver no ponto central, a busca termina com sucesso (o elemento foi encontrado). Caso contrário, deve-se continuar a busca usando somente uma das duas partes do vetor: a parte à esquerda contém os elementos menores que o ponto central, e a parte à direita contém os elementos maiores. Note que isto só é possível porque assume-se que a lista está ordenada.*

Neste exercício, você receberá uma lista com uma quantidade de elementos e realizará uma única busca binária procurando por um elemento. Seu propósito será contabilizar a quantidade de comparações realizadas na busca binária.

Note que, se a quantidade de elementos da lista for **ímpar** em qualquer etapa da busca, haverá um elemento central dividindo-a em duas partes de tamanhos iguais. Contudo, se a quantidade de elementos for **par**, o elemento central estará em uma das partes (ele pode ser o último elemento da primeira parte, ou o primeiro elemento da segunda parte). Neste exercício, sempre que a quantidade de elementos for **par** (em qualquer etapa da busca), **convencionaremos** que você deverá usar o **primeiro elemento da segunda parte** como sendo o elemento central. Segue exemplo: considere que o índice do primeiro elemento é 0, como usado pelas listas de python. Para uma quantidade de **n** elementos, se o **tamanho** da lista **n** for **par**, o índice do elemento central será o elemento de **índice n/2** (primeiro elemento da segunda metade). Se **n** for **ímpar**, o índice do elemento central será o elemento de **índice piso(n/2)** (obs.: a função piso em python é **floor** e está na biblioteca **math**, você pode importá-la **antes** de utilizá-la com o comando: **from math import floor**).

Crie um programa em python 3 para tratar as entradas e saídas conforme instruções a seguir.

Regra Geral: você deverá criar uma função **recursiva** para realizar a busca. Não é permitido o uso de *loops* nas etapas da busca binária. Isto é, somente será permitido usar *loops* para processar os elementos da entrada.

Entrada

O arquivo de entrada consiste de um número a ser consultado (buscado) na lista. Em seguida, vários **números (IDs)** que **compõem a lista** (considere que os elementos da lista sempre estarão ordenados). O número para consulta deve ser procurado na lista a partir do nó central. Considere que não haverá repetição de elementos. Os elementos da entrada serão separados por **espaço** conforme exemplo a seguir.

Saída

A saída deve conter apenas o resultado da procura pelo primeiro número da entrada. Portanto, deverá imprimir a **quantidade de comparações de elementos** realizadas até que ele seja encontrado ou até atingir o fim da busca sem encontrá-lo (caso não exista na lista). A saída deve ter a quantidade de comparações.

Exemplo

Entrada1:	Saída1:
5 **1 2 3 4 5 6 7**	3

Obs: na entrada do exemplo, 5 é o número para realizar a busca que deve ser computado. Em seguida, aparecem sete elementos (IDs, destacados em negrito somente para melhor visualização). A saída deverá ser a quantidade de comparações da busca do número 5 na lista (total: 3 comparações).

Árvore representando o exemplo da entrada1: