1)

Search in Rotated Array: Given a sorted array of n integers that has been rotated an unknown number of times, write code to find an element in the array. You may assume that the array was originally sorted in increasing order.

EXAMPLE

Input: find 5 in {15, 16, 19, 20, 25, 1, 3, 4, 5, 7, 10, 14}

Output: 8 (the index of 5 in the array)

Assuma que a lista nao contenha numeros repetidos. Obtenha um algoritmo com complexidade assintotica estritamente melhor que O(n) (e.g., O((log n)^2) ou O(log n)). Analise seu algoritmo.

Dica: Note que uma das metades do vetor esta ordenado.

Solucao:

2)

Smallest Difference: Given two arrays of integers, compute the pair of values (one value in each array) with the smallest (non-negative) difference. Return the difference.

EXAMPLE

Input: {1, 3, 15, 11, 2}, {23, 127, 235, 19, 8}

Output: 3. That is, the pair (11, 8).

Resolva esse problema em tempo O(n log n) (onde os dois vetores tem tamanho no maximo n)

Solucao:

Given two sorted lists of numbers of length m, n. Give an algorithm that finds the k'th smallest number in the union of the lists, in time $O(\log m + \log n)$

(You can assume that all numbers in the input are distinct).

Dica: Em cada iteracao voce pode descartar a metade de uma das listas. Imagine a posicao dos elementos do meio das listas na uniao ordenada das listas.

Solucao: