

## Kata 0

Given an array of integers, return indices of the two numbers such that they add up to a specific target. You may assume that each input would have exactly one solution, and you may not use the same element twice.

### Example:

Given nums = [2, 7, 11, 15], target = 9,  
Because nums[0] + nums[1] = 2 + 7 = 9,  
return [0, 1].

## Breakdown

- ¿Qué si no dan? No hay dos números t.q. su suma no obtenga el resultado.
- ¿Está ordenado? Puede impactar el desempeño.
- Sólo se admiten números positivos  $> 0$ . No hay negativos.
- Tamaño del arreglo:  $n \geq 2$

### → El caso más sencillo.

$n=2$ , respuestas triviales  $\rightarrow$  existe o no.

### → Caso $n=3$ $A = [2, 7, 4]$

\* Se busca el índice, no el valor.

$i = 0$

$A[0] = 2$

$target - A[0] = \text{diferencia}$  // Para A, se busca  $6 - 2 = 4$

// Algoritmo 1

$i = 0$

Seleccionar elemento // ciclo en A.

Buscar en el resto // binary search, por ejemplo.

### Análisis

$n$  ( $n \log n$  + ordenar)  $\approx n^2$   
recorrer A  $\rightarrow$  búsqueda  $\rightarrow$  Quicksort

Dado que se tiene que recorrer el arreglo, la complejidad mínima es de  $n$ .

## Segunda Alternativa - Hash Tables

Hash tables tienen búsqueda en  $O(1)$  para casos promedio.

inicial =  $A[0]$

diff = target - inicial

Insertar en HTable

búsqueda en HTable

return (index, value) // Exito

seguir buscando

indice++

### Análisis

Caso promedio:  $n + n = 2n \approx n$

Peor caso:  $n(n) + n(n) = 2n^2 \approx n^2$