

PROGRAMACIÓN II: GUIA 01

Prof. Dr. Hans H. Ccacyahuillca Bejar, INFO-UNSAAC

06/09/2023

Problema 1

Implementar el algoritmo de Fibonacci en su forma no recursiva y recursiva:

Algorithm 1: $F(n)$

Input: Some non-negative integer n
Output: The n th number in the Fibonacci Sequence

```
if  $n \leq 1$  then
  | return  $n$ 
else
  | return  $F(n-1) + F(n-2)$ ;
```

Problema 2

Se le proporciona un número entero grande representado como un arreglo de dígitos enteros, donde cada dígito $[i]$ es el i ésimo dígito del número entero. Los dígitos están ordenados del más al menos significativo, de izquierda a derecha. El número entero grande no contiene ceros a la izquierda.

Incrementa el número entero grande en uno y devuelve el arreglo de dígitos resultante.

Ejemplo 1:

Entrada: $\text{digitos} = [1, 2, 3]$

Salida: $[1, 2, 4]$

Explicación: El arreglo representa el entero 123.

Al incrementar 1 da como resultado $123 + 1 = 124$.

Entonces, el resultado es $[1, 2, 4]$.

Ejemplo 2:

Entrada: $\text{digitos} = [9]$

Salida: $[1, 0]$

Explicación: El arreglo representa el entero 9.

Al incrementar 1 da como resultado $9 + 1 = 10$.

Entonces, el resultado es $[1, 0]$.

Restricciones:

- $1 \leq \text{digits.length} \leq 100$
- $0 \leq \text{digits}[i] \leq 9$
- el arreglo `digits` no contiene ningún cero a la izquierda (no comienza con cero).

Problema 3

Dada un arreglo de números de tamaño n , devolver el elemento mayoritario.

El elemento mayoritario es el elemento que aparece más de $\lfloor n/2 \rfloor$ veces. Puede suponer que el elemento mayoritario siempre existe en el arreglo.

Ejemplo 1:

Entrada: `nums = [3, 2, 3]`

Salida: 3

Ejemplo 2:

Entrada: `nums = [2, 2, 1, 1, 1, 2, 2]`

Salida: 2

Restricciones:

- $n == \text{nums.length}$
- $1 \leq n \leq 5 * 10^4$
- $-10^9 \leq \text{nums}[i] \leq 10^9$