# Ayudantía 9

Decrecer y Conquistar

Universidad Técnico Federico Santa Maria

*Carlos Lagos* *carlos.lagosc@usm.cl*
*Nangel Coello* *nangel.coello@usm.cl*

22 de junio de 2024

# Contenidos

# Fibonacci

## Enunciado

La sucesión de Fibonacci se puede representar de manera matricial de la siguiente forma:

$$\begin{pmatrix} f_{n-1} & f_n \\ f_n & f_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n$$

Ejemplo, si queremos representar el cálculo de $f_5$:

$$\begin{pmatrix} f_4 & f_5 \\ f_5 & f_6 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^5 = \begin{pmatrix} 3 & 5 \\ 5 & 8 \end{pmatrix}$$

Entonces, $f_5 = 5$.
Comparado con la programación dinámica, no muestra una ventaja clara.
Puede ser un poco más costoso, ya que implica elevar una matriz $2 \times 2$.
Por eso, se solicita diseñar un algoritmo de decrecer y conquistar que resuelva este problema en $O(\log(n))$.

# GCD y LCM

## Enunciado

Dado dos arreglos $A[1...n]$ y $B[1...n]$, ambos de tamaño $n$, queremos calcular $\sum_{i=1}^{n} \text{lcm}(A_i, B_i) - \text{gcd}(A_i, B_i)$ para $i$ desde $1$ hasta $n$. Diseña un algoritmo que resuelva lo anterior en $O(n \log(min(A_{\text{max}}, B_{\text{max}})))$.

## Definiciones

$gcd(a, b)$ es el máximo común divisor entre $a$ y $b$, mientras que $lcm(a, b)$ es el mínimo común múltiplo entre $a$ y $b$.

# GCD y LCM

## Enunciado

Dado dos arreglos $A[1...n]$ y $B[1...n]$, ambos de tamaño $n$, queremos calcular $\sum_{i=1}^{n} \text{lcm}(A_i, B_i) - \text{gcd}(A_i, B_i)$ para $i$ desde $1$ hasta $n$. Diseña un algoritmo que resuelva lo anterior en $O(n \log(min(A_{\text{max}}, B_{\text{max}})))$.

## Definiciones

$gcd(a, b)$ es el máximo común divisor entre $a$ y $b$, mientras que $lcm(a, b)$ es el mínimo común múltiplo entre $a$ y $b$.

## Pistas

1. Si $a = bd + r$, entonces $gcd(a, b) = gcd(b, r)$.

# GCD y LCM

## Enunciado

Dado dos arreglos $A[1...n]$ y $B[1...n]$, ambos de tamaño $n$, queremos calcular $\sum_{i=1}^{n} \mathrm{lcm}(A_i, B_i) - \mathrm{gcd}(A_i, B_i)$ para $i$ desde 1 hasta $n$. Diseña un algoritmo que resuelva lo anterior en $O(n \log(min(A_{\mathsf{max}}, B_{\mathsf{max}})))$.

## Definiciones

$gcd(a, b)$ es el máximo común divisor entre $a$ y $b$, mientras que $lcm(a, b)$ es el mínimo común múltiplo entre $a$ y $b$.

## Pistas

1. Si $a = bd + r$, entonces $gcd(a, b) = gcd(b, r)$.
2. $lcm(a, b) = \frac{ab}{gcd(a,b)}$

# Arreglo shifteado

## Enunciado

Suponga que se tiene un arreglo ordenado $A[0..n-1]$ de enteros, en donde los elementos han sido movidos ("shifteados") $k$ posiciones a la derecha, de manera circular. Por ejemplo, $A = \langle 35, 42, 5, 15, 27, 29 \rangle$ es un arreglo ordenado que ha sido shifteado circularmente $k = 2$ posiciones, mientras que $A = \langle 27, 29, 35, 42, 5, 15 \rangle$ ha sido shifteado $k = 4$ posiciones.

## Ejercicios

a) Suponga que se conoce $k$. Dar un algoritmo de tiempo $\Theta(1)$ que permita encontrar el mayor valor almacenado en $A$.
b) Suponga que no se conoce el valor de $k$. Dar un algoritmo que permita encontrar el mayor elemento almacenado en $A$ en tiempo $O(\log n)$.
c) Suponga que no se conoce el valor de $k$. Dar un algoritmo que permita encontrar cualquier elemento almacenado en $A$ en tiempo $O(\log n)$.