



## Sesión 3: Divide y vencerás I

Marzo 22, 2020

Lee cuidadosamente cada uno de los siguientes ejercicios de programación y haz lo que se te pide. Aunque está permitido que discutas las soluciones con tus compañeros, **debes desarrollar tus propios algoritmos y escribir tu propio código fuente**. Si se detecta que copiaste código fuente, tu calificación puede ser menor a 5.

### 1. Diseño de algoritmos

1. Analiza la siguiente regla para calcular el máximo común divisor

$$\text{mcd}(a, b) = \begin{cases} 2\text{mcd}(a/2, b/2) & \text{si } a, b \text{ son pares} \\ \text{mcd}(a, b/2) & \text{si } a \text{ es impar, } b \text{ es par} \\ \text{mcd}((a - b)/2, b) & \text{si } a, b \text{ son impares} \end{cases}$$

- a) ¿Es correcta? es decir, al utilizarla ¿es posible calcular correctamente el mcd? Justifica tu respuestas
  - b) Basándote en ella, escribe el pseudocódigo de un algoritmo que utilice la estrategia de divide y vencerás para calcula el mcd.
  - c) Calcula la complejidad de dicho algoritmo
2. Diseña un algoritmo que dado un arreglo con los coeficientes de un polinomio  $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$  y un valor  $a$ , devuelva la evaluación del polinomio en  $a$ , es decir, el valor de  $p(a)$ . Observa que el algoritmo más trivial tendrá complejidad  $O(n^2)$ , diseña un algoritmo que utilice la estrategia de divide y vencerás. Calcula la complejidad de tu algoritmo.
  3. Dado un arreglo  $A$  de  $n$  enteros distintos, se quiere contar el número de pares de índices  $(i, j)$  tales que  $i < j$  y  $A[i] > 2A[j]$ .
    - Diseña un algoritmo por fuerza bruta para solucionar el problema anterior. Realiza el análisis correspondiente para determinar la complejidad de este algoritmo.
    - Diseña una segunda versión para solucionar el problema, cuya complejidad sea  $O(n \log n)$ .

En ambos casos, muestra con un ejemplo el funcionamiento de tus algoritmos.

## 2. Ejercicios de programación.

1. Elige C/C++, Java o Python e implementa cada uno de los algoritmos de la sección anterior.

## 3. Productos a entregar

Debes escribir un reporte por equipo que contenga:

1. Datos que te identifiquen, así como la fecha y título de la practica
2. Escaneo o foto de los ejercicios resueltos en tu cuaderno, para la sección 1.
3. Las partes más importantes de tu código, junto con una breve explicación que ayude a entender tu código.
4. Incluye un ejemplo para cada algoritmo, mostrando que tu algoritmo funciona.
5. Impresiones de pantalla que muestren el correcto funcionamiento de tu programa.

**Este reporte deberás subirlo a google classroom el próximo lunes 30 de marzo.** En la carpeta de google drive exclusiva para el curso de Análisis de Algoritmos, agrega un zip con tu código fuente de cada ejercicio. El nombre del archivo debe estar conformado por tus apellidos completos, seguidos del sufijo lab03AA. Por ejemplo, Andrés Zarate Cano, deberá nombrar al archivo *Zarate-Cano\_lab03AA.zip*. La fecha límite para hacerlo es el 30 de marzo.