

1. Descarga la clase AnalisisAlgoritmos.java que se encuentra en Blackboard. Revisa su funcionamiento y discute en equipo su objetivo.

- ¿Para qué sirve la variable CANTIDAD\_ELEMENTOS?
- ¿Cuál es el objetivo del método nanoTime( ) de la clase System?
- ¿Qué imprime el programa?

2. Diseña un algoritmo que se encargue de instanciar y llenar un arreglo de enteros de tamaño size con valores enteros aleatorios. Incluye números negativos también. Prográmalo con la siguiente firma: `public static int[] refreshArray(int size)`

Ejemplo: `refreshArray(10)` → 

4	1	2	-3	1	8	5	3	-2	10
---	---	---	----	---	---	---	---	----	----

3. Crea una implementación del algoritmo BubbleSort

4. Crea una implementación del algoritmo MergeSort.

5. Diseña un algoritmo BogoSort con la firma `public static void BogoSort(int[] array)` que realice las siguientes operaciones:

- Verifique si el arreglo está ordenado (de menor a mayor). De ser así, termine la ejecución del método.
- Genera dos índices aleatorios e intercambia los elementos en dichas posiciones.
- Verifique si arreglo está ordenado. De ser así, termine la ejecución.
- Si el arreglo no está ordenado, continúa a partir del paso ii.

6. Modifica la clase AnalisisAlgoritmos para que permita evaluar cada uno de los tres algoritmos de ordenamiento creados en los puntos 3, 4 y 5. Genera un programa que permita capturar el algoritmo a utilizar y la cantidad de elementos a ordenar.

**PUNTOS EXTRAS:** ¿Puedes automatizar la generación de la tabla completa de resultados? Genera los métodos necesarios.

Algoritmo ▾	# Elementos ▾	Ejecución 1 ▾	Ejecución 2 ▾	Ejecución 3 ▾	Ejecución 4 ▾	Ejecución 5 ▾	Promedio ▾
Bubble Sort	100						
Bubble Sort	1,000						
Bubble Sort	10,000						
Bubble Sort	100,000						
Bubble Sort	200,000						
Merge Sort	100						
Merge Sort	1,000						
Merge Sort	10,000						
Merge Sort	100,000						
Merge Sort	1,000,000						
Bogo Sort	3						
Bogo Sort	6						
Bogo Sort	9						
Bogo Sort	12						
Bogo Sort	15						

7. Llena la tabla resultados.xlsx con los resultados de cada una de las ejecuciones obtenidas.

8. Grafica los resultados de la columna “Promedio” para cada uno de los algoritmos. Utiliza alguna herramienta de software como Excel, Wolfram Alpha, Matlab, etc.

9. Guarda las gráficas en un archivo de Word. Adicionalmente en el documento, responde las siguientes preguntas:

- ¿Cuál es el algoritmo de ordenamiento más rápido? Utiliza la notación Big-O para justificar tu respuesta.
- ¿Cuál es el algoritmo de ordenamiento más lento? Utiliza la notación Big-O para justificar tu respuesta.
- ¿Las gráficas se aproximan a la notación Big-O esperada?
- ¿Qué factores (de hardware, software, externos) existen que afecten los tiempos de ordenamiento entre distintas ejecuciones? Menciona por lo menos 5.