INFORME DE LA INGENIERIA INVERSA

CLASES IDENTIFICADAS EN EL JAR:

BubleSort.class

MergeSortDemo.class

QuickSort.class

SumArray.class

App.class

OPERACIONES SOBRE ARREGLOS IDENTIFICADAS:

- Uso de arreglos de tipo int[]
- Acceso a elementos mediante índices
- Recorrido de arreglos mediante ciclos for y for-each
- Comparación de valores del arreglo
- Intercambio de elementos dentro del arreglo
- Copia de subarreglos utilizando la clase Arrays
- Cálculo de la suma de los elementos del arreglo
- Ordenamiento de arreglos mediante:
    - Bubble Sort
    - Merge Sort
    - Quick Sort
- Método Arrays.sort()
- Copia y manejo de subarreglos usando la clase Arrays
- de la suma de los elementos del arreglo

Qué algoritmos de ordenamiento se utilizan:

Algoritmos de ordenamiento utilizados:

- Bubble Sort: ordena comparando elementos adyacentes e intercambiándolos si están en el orden incorrecto.
- Merge Sort: ordena dividiendo el arreglo en partes más pequeñas y luego las combina ordenadamente.
- Quick Sort: ordena usando un pivote y particionando el arreglo de forma recursiva.

- Sort de la clase Arrays: utiliza el método Arrays.sort() para ordenar arreglos automáticamente.

- **SumArray.class**
  Permite calcular la suma de todos los elementos de un arreglo de enteros.
- **App.class**
  Clase principal que prueba el funcionamiento de las demás clases utilizando arreglos y mostrando resultados en consola.

PARTE 2: Segundo número mayor y segundo número menor

Algoritmo paso a paso

Pseudocodigo

1. Inicializar variables:
   mayor = -∞
   segundoMayor -∞
   menor = -∞
   segundoMenor = -∞
2. Para cada numero n en el arreglo
   a) Comparar con el mayor:
      Si n > mayor;
        segundoMayor = mayor
        mayor = n
      Sino si n > segundoMayor;
        segundoMayor = n
   b) Comparar con el menor:
      Si n < menor:
        segundoMenor = menor
        menor = n
      Sino si n < segundoMenor:
        SegundoMenor = n
3. Al finalizar todo el recorrido
   Imprime “Segundo mayor: “ , segundoMayor
   Imprime “Segundo menor: “ , segundoMenor

Explicación Breve

Mientras recorremos el arreglo solo una vez, actualizamos cuatro variables para mantener los dos números más grandes y los dos más pequeños. Así, al final, tenemos el segundo mayor y el segundo menor sin ordenar el arreglo.

Análisis de complejidad:

Tiempo: O(n) → recorremos el arreglo solo una vez

Espacio: O(1) → usamos solo cuatro variables