



**Tecnológico
de Monterrey**

Instituto Tecnológico de Estudios Superiores Monterrey

CAMPUS QUERÉTARO

Análisis y diseño de algoritmos avanzados

Ramona Fuénte Valdéz

Grupo 602

**Actividad 1.4. Implementación de la técnica de programación "divide y
vencerás"**

PRESENTAN

Rodrigo Terán Hernández

A01704108

Fecha:
27/08/2023

Captura de pantalla de la ejecución del programa

Test 1

```
● rodrigoteranhernandez@MacBook-Pro-de-Rodrigo A14_MergeSort_Individual % sh scripts/run_tests.sh
7
38
27
43
3
9
82
10
Sorted:
82
43
38
27
10
9
3
○ rodrigoteranhernandez@MacBook-Pro-de-Rodrigo A14_MergeSort_Individual %
```

Test 2

```
● rodrigoteranhernandez@MacBook-Pro-de-Rodrigo A14_MergeSort_Individual % sh scripts/run_tests.sh
10
5
2
7
6
2
1
0
3
9
4
Sorted:
9
7
6
5
4
3
2
2
1
0
○ rodrigoteranhernandez@MacBook-Pro-de-Rodrigo A14_MergeSort_Individual %
```

Test 3

- Para este test decidí incluir números decimales y números negativos
- También incluí el 0

```
● rodrigoteranhernandez@MacBook-Pro-de-Rodrigo A14_MergeSort_Individual % sh scripts/run_tests.sh
5
-4.3
24.3
1.1
23.3
0
Sorted:
24.3
23.3
1.1
0
-4.3
○ rodrigoteranhernandez@MacBook-Pro-de-Rodrigo A14_MergeSort_Individual %
```

Test 4

- En este test incluí más números
- Una mezcla de enteros y decimales

```
● rodrigoteranhernandez@MacBook-Pro-de-Rodrigo A14_MergeSort_Individual % sh scripts/run_tests.sh
11
3.23
324.5
64
33
7.7
31
727
2.63
52.6
183.58
2
Sorted:
727
324.5
183.58
64
52.6
33
31
7.7
3.23
2.63
2
○ rodrigoteranhernandez@MacBook-Pro-de-Rodrigo A14_MergeSort_Individual %
```

Complejidad $O(n \log(n))$

Este algoritmo tiene una complejidad de $O(n \log(n))$ ya que cuando dividimos la lista de números a ordenar estamos creando un árbol binario con una altura de $\log(n)$ y en cada nivel vamos a tener que correr el algoritmo de merge ya que necesitamos juntar las dos partes ya ordenadas. Ese algoritmo toma $O(n)$ y como lo vamos a hacer $\log(n)$ veces, entonces al final nos da una complejidad de tiempo $O(n \log(n))$.

¿Por qué es “divide y vencerás”?

Este algoritmo utiliza la estrategia “divide y vencerás” ya que al momento de dividir la lista de números a la mitad y llamarse a sí mismo con cada una de esas mitades estamos reciclando la funcionalidad. Es decir, estamos dividiendo el trabajo y aprovechando el código ya escrito. Y al final reconstruyes todas esas sub respuestas para construir la respuesta al problema original.