

## Laboratorio Nro. X

### Escribir el tema del laboratorio

**Andrés Ospina Patiño**  
Universidad Eafit  
Medellín, Colombia  
aospinap1@eafit.edu.co

**Felipe Álvarez Benítez**  
Universidad Eafit  
Medellín, Colombia  
falvarezb@eafit.edu.co

### 3) Simulacro de preguntas de sustentación de Proyectos

#### 3.1

InsertionSort	
Tiempo	Tamaño
0	0
293	435
10	870
12	1305
68	1740
10	2175
3	2610
13	3045
77	3480
12	3915
37	4350
110	4785
18	5220
20	5655
39	6090
73	6525
175	6960
103	7395
62	7830
52	8265
MargeSort	
Tiempo	Tamaño

**PhD. Mauricio Toro Bermúdez**  
Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

## ESTRUCTURA DE DATOS 1

### Código ST0245

0	0
2	435
1	870
0	1305
5	1740
2	2175
72	2610
27	3045
12	3480
26	3915
2	4350
25	4785
1	5220
2	5655
2	6090
2	6525
2	6960
2	7395
3	7830
2	8265

### 3.2

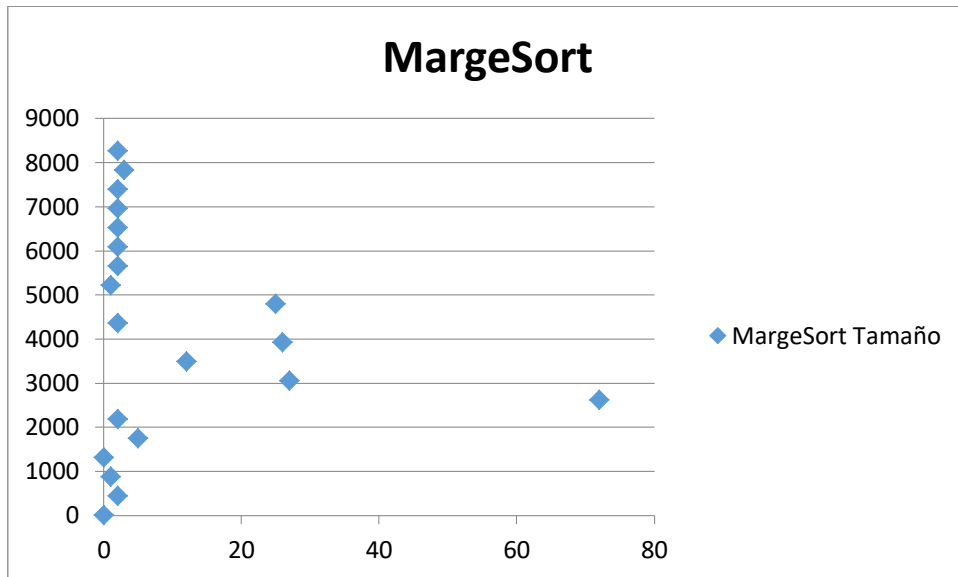


**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
 Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
 Tel: (+57) (4) 261 95 00 Ext. 9473

## ESTRUCTURA DE DATOS 1

### Código ST0245



**3.3** MergeSort es mucho más eficiente que el InsertionSort para arreglos grandes, al comparar la complejidad asintótica.

**3.4** Teniendo en cuenta la complejidad asintótica de ambos métodos, InsertionSort  $O(2^n)$  y MergeSort  $O(n \log n)$ , se puede ver claramente que el segundo de ellos es más eficiente, además el MergeSort maneja una gran cantidad de números y recurre a procesos más óptimos, haciendo que el método se demore menos en ser ejecutado.

**3.5** No, no es el más eficiente, mientras que el InsertionSort tiene una complejidad de  $O(2^n)$  el MergeSort tiene una complejidad de  $O(n \log n)$ .

**3.6** MaxSpan consiste en ver cuantos elementos hay entre el primer y el último elemento del arreglo, incluidos estos dos pero considerados como uno, a menos que el primer y último elemento sean iguales, en ese caso se toman en cuenta los dos. Para resolverlo solo fue necesario plasmar todas estas condiciones.

### 3.7

#### Array 2:

- countEvents:  $O(n)$ .
- bigDiff:  $O(n)$ .
- centeredAverage:  $O(n)$ .
- sum13:  $O(n)$ .
- fizzBuzz:  $O(n)$ .

#### Array 3:

- maxSpan:  $O(1)$ .
- fix34:  $O(n)$ .
- fix45:  $O(n^2)$ .
- canBalance:  $O(n^2)$ .
- seriesUp:  $O(n^2)$ .

**3.7** La variable  $n$  representa el número de elementos de cada uno de los arreglos.

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
 Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
 Tel: (+57) (4) 261 95 00 Ext. 9473

**ESTRUCTURA DE DATOS 1**  
**Código ST0245**

**4) Simulacro de Parcial**

**4.1** c

**4.2** d

**4.3** b

**4.4** b

**4.5** d

**4.6** a

**4.7**

-  $T(n)=T(n-1)+c$

-  $O(n)$

**4.8** a

**4.9** c

**4.10** c

**4.11** c

**4.12** a

**4.13** c

**4.14** b

**5) Lectura recomendada (opcional)**

Mapa conceptual

**6) Trabajo en Equipo y Progreso Gradual (Opcional)**

**6.1** Actas de reunión

**6.2** El reporte de cambios en el código

**6.3** El reporte de cambios del informe de laboratorio

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473