

## Laboratorio Nro. 2

### Escribir el tema del laboratorio

**Manuela Herrera López**  
Universidad Eafit  
Medellín, Colombia  
mherreral@eafit.edu.co

**Samuel David Palacios Bernate**  
Universidad Eafit  
Medellín, Colombia  
sdpalaciob@eafit.edu.co

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

#### 3.1 Tiempos para InsertionSort

**N      Tiempo**

1000	16397100
1001	19610200
1002	22099400
1003	25443400
1004	28829300
1005	30084500
1006	31176600
1007	32285500
1008	42418000
1009	45197000
1010	47891500
1011	65150500
1012	67536000
1013	68659600
1014	69720600
1015	80516600
1016	87350700
1017	88634800
1018	89715100
1019	90794700

**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**

**Tiempos para MergeSort**

<b>N</b>	<b>Tiempo</b>
1000	5359500
1001	12965000
1002	16648900
1003	19939000
1004	21439100
1005	22341700
1006	27730500
1007	28686000
1008	30718200
1009	31592500
1010	36072700
1011	36928400
1012	37686500
1013	42358900
1014	44312500
1015	45313600
1016	46124200
1017	47649700
1018	48512000
1019	49622800

**PhD. Mauricio Toro Bermúdez**

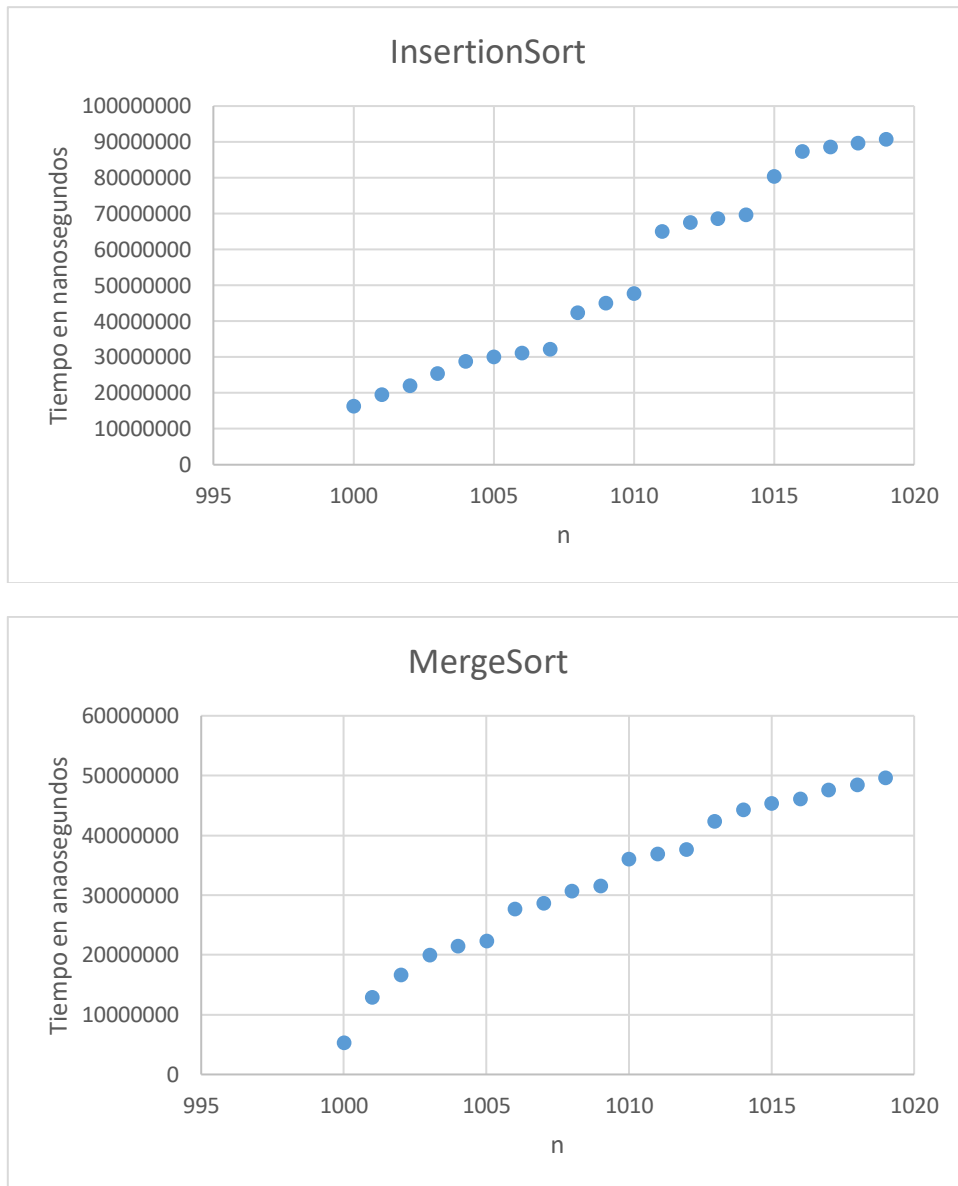
Docente | Escuela de Ingeniería | Informática y Sistemas  
 Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
 Tel: (+57) (4) 261 95 00 Ext. 9473



## ESTRUCTURA DE DATOS 1

### Código ST0245

### 3.2



**3.3** MergeSort es un algoritmo mucho más eficiente que InsertionSort, ya que para valores grandes, que es cuando se ve la diferencia en tiempos, este es menor, ya que no hace (en el peor de los casos)  $n$  veces la comparación, sino que lo hace de  $2 * \log 2$ , lo que hace realmente una gran diferencia.

**3.4** No sería apropiado Insertion para un videojuego con millones de elementos, porque se tardaría una infinidad de tiempo en la comparación de todos los elementos del arreglo.

**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

**3.5** Para que insertionSort sea más rápido, el arreglo debe estar casi ordenado, para que cuando se haga la llamada al Insertion sean pocos los números que se deban correr una posición, o bien, que en la llamada ningún elemento se deba mover de posición.

### 3.6

El método maxSpan busca encontrar la subsecuencia de números entre dos valores iguales considerando únicamente los valores más a la izquierda y más a la derecha. Para lograrlo primero se evalúa si el arreglo está vacío, de ser así retornará el valor del tamaño del arreglo, es decir, cero (0). En caso de que el arreglo tenga un tamaño mayor que cero se evaluará si la primera posición (Más a la izquierda), es diferente de la última posición (Más a la derecha), entonces se dice que la subsecuencia es igual al tamaño del arreglo menos 1, de no ser así se dice que la subsecuencia será igual al tamaño del arreglo.

**3.7** Los 5 ejercicios realizados de la sección 2.1 se expresan con una complejidad de  $O(n)$

**3.8** “n” indica la cantidad de pasos para terminar la ejecución del programa en el peor de los casos. Se dice que es de orden lineal ya que ejecuta una secuencia de pasos “n” veces.

## 4) Simulacro de Parcial

**4.1**  $D \rightarrow O(1)$

**4.2**  $B \rightarrow O(m \cdot n \cdot \sqrt{n})$

**4.3**  $B \rightarrow O(\text{ancho})$

**4.4**  $A \rightarrow O(n^4)$

**4.5**  $D \rightarrow O(n^2)$

**4.6**  $A \rightarrow T(n) = T(n - 1) + c$

**4.7** Q

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

4.7.2  $O(n)$

**4.8** A

**4.9** D

**4.10** Error en el enunciado del problema

**4.11** C

**4.12** B

**4.13** C

**4.14** A

**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