

## Práctica de laboratorio N°2: Complejidad

**Juan Andrés Arroyave Sánchez**  
Universidad Eafit  
Medellín, Colombia  
jaarroyavs@eafit.edu.co

**Gustavo Adolfo López García**  
Universidad Eafit  
Medellín, Colombia  
galopezg@eafit.edu.co

### 3) Practice for final project defense presentation

#### 3.1

Cantidad de datos	Tiempo de Ejecución
10000	25900662
20000	140106382
30000	183968935
40000	307499699
50000	484024086
60000	674854064
70000	924492977
80000	1207728757
90000	1524267081
100000	1978870106
110000	2413276376
120000	2698774675
130000	3171624037
140000	3666754502
150000	4213591082

**PhD. Mauricio Toro Bermúdez**  
Professor | School of Engineering | Informatics and Systems  
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627  
Phone: (+57) (4) 261 95 00 Ext. 9473

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

<b>160000</b>	4809566270
<b>170000</b>	5405887925
<b>180000</b>	6055510167
<b>190000</b>	6798793323
<b>200000</b>	8083919474
<b>210000</b>	8287693660

<b>Cantidad de datos</b>	<b>Tiempo de ejecución</b>
<b>10000</b>	4807905
<b>20000</b>	5943313
<b>30000</b>	8227167
<b>40000</b>	11141935
<b>50000</b>	12602084
<b>60000</b>	27131675
<b>70000</b>	21905005
<b>80000</b>	27578491
<b>90000</b>	19527521
<b>100000</b>	25347181
<b>110000</b>	26723181
<b>120000</b>	29475579
<b>130000</b>	42306404
<b>140000</b>	38240030
<b>150000</b>	40737612
<b>160000</b>	48072334

**PhD. Mauricio Toro Bermúdez**

Professor | School of Engineering | Informatics and Systems

Email: mtorobe@eafit.edu.co | Office: Building 19 – 627

Phone: (+57) (4) 261 95 00 Ext. 9473



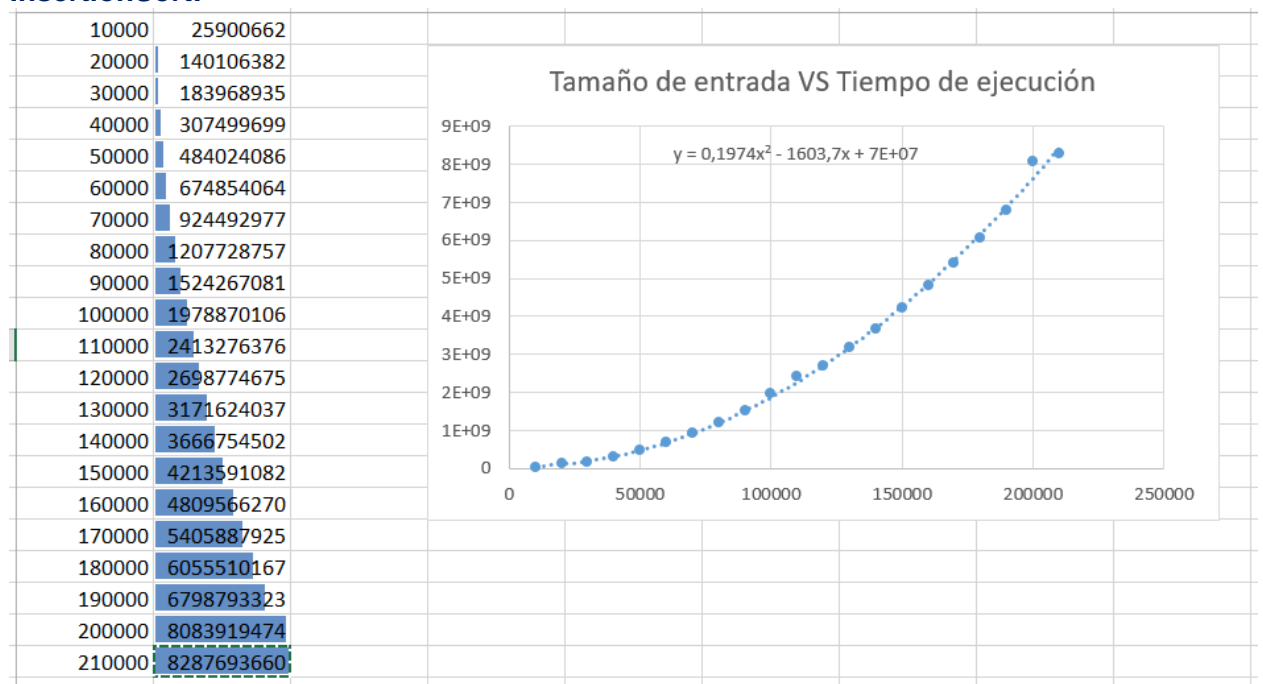
## ESTRUCTURA DE DATOS 1

### Código ST0245

170000	45406851
180000	51549670
190000	48486359
200000	49918459
210000	56363501

### 3.2

#### InsertionSort:



#### MergeSort:

**PhD. Mauricio Toro Bermúdez**

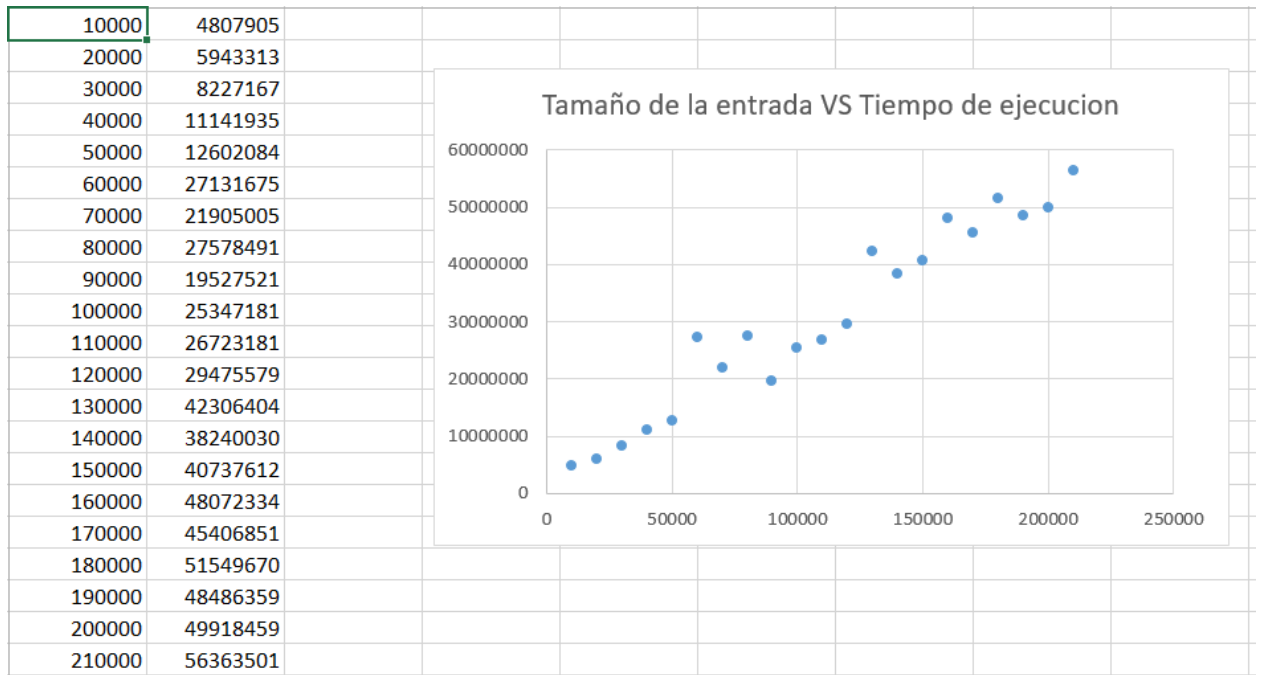
Professor | School of Engineering | Informatics and Systems

Email: mtorobe@eafit.edu.co | Office: Building 19 – 627

Phone: (+57) (4) 261 95 00 Ext. 9473

## ESTRUCTURA DE DATOS 1

### Código ST0245



**3.3** A medida que los datos aumentan se quiere que el ordenamiento sea lo más rápido posible por lo cual al usar el método de Insertion Sort desfallece en estos casos. Por lo tanto es más recomendable el uso de Merge Sort.

**3.4** No es apropiado usarlo para un videojuego que posee millones de elementos ya que a medida que la cantidad de datos aumenta el tiempo que demora la ejecución se incrementa a una gran velocidad por lo cual no es recomendable usarlo en estos casos.

**3.5** Insertion sort es mejor en arreglos de 1000 o menos datos ya que al sobrepasar este punto merge sort ordenara los datos a una mayor velocidad que es lo que se busca en estos casos particulares.

**3.6** El ejercicio maxSpan tiene la siguiente función, primero nos pasan un arreglo de  $n$  elementos como parámetro, el problema que tenemos que resolver es el de hallar la longitud de un bloque en el arreglo dado, con una condición, los extremos de los valores serán iguales, por ejemplo, si tenemos el arreglo 1,2,2,1 lo que debe retornar el programa es 4.

Para resolver este problema lo que hicimos fue lo siguiente, creamos una variable llamada span, que será la variable que retornaremos al final, luego de esto recorremos por primera vez el ciclo, esto tendrá la funcionalidad de obtener cada posición en nuestro arreglo, continuando haremos otro ciclo que vaya desde la última posición hasta la primera y compararemos si hay un elemento en una posición  $j$  que es igual a un elemento en una posición  $i$ , esto porque si los comparamos al revés habrá un `ArrayIndexOutOfBoundsException`, esta relación en matemáticas es anti reflexiva.

Siguiendo: Entonces dada la condición le daremos el valor de la posición  $j$  menos la  $i$ -ésima posición +1, esto con la finalidad de saber que longitud hay. ¿Está claro hasta ahí? Finalmente si esa variable es mayor que span, que comienza en 1 por conveniencia ya que si hay al menos dos valores iguales la longitud será uno, ese valor tomará span.

**PhD. Mauricio Toro Bermúdez**

Professor | School of Engineering | Informatics and Systems

Email: mtorobe@eafit.edu.co | Office: Building 19 – 627

Phone: (+57) (4) 261 95 00 Ext. 9473

Finalmente, si el arreglo es vacío span será 0, porque no hay elementos que son iguales.

#### 4) Practice for midterms

- 4.1** Sabemos que la complejidad de  $P(m, n)$  es mayor que la de  $H(n, m)$ , por lo que la complejidad definitivamente no podría ser parte de la CONDICIÓN dada que  $P(n, m)$  crece más rápidamente es  $O(n+m)$ .
- 4.2** B)  $O(m \cdot n \cdot \text{raíz de } n)$ .
- 4.3** B)  $O(\text{Ancho})$ .
- 4.4** B)  $O(N^3)$ .
- 4.5** D)  $O(N^2)$ .
- 4.6** D)  $T(n) = T(n+1) + C$ .
- 4.7** Respuestas:
- 4.7.1**  $T(n) = T(n) + T(n-1) + C$ .
- 4.7.2**  $O(N)$ .
- 4.8** D) La Función misterio(n) ejecuta  $O(n)$  pasos.
- 4.9** C) Ejecuta  $n \cdot m$  pasos.
- 4.10** –
- 4.11** Ejecuta  $T(n) = T(n-1) + T(n-2) + C$  pasos.
- 4.12** Si.
- 4.13** –
- 4.14** B) Ejecuta  $O(n \cdot m)$  instrucciones.

**PhD. Mauricio Toro Bermúdez**

Professor | School of Engineering | Informatics and Systems

Email: mtorobe@eafit.edu.co | Office: Building 19 – 627

Phone: (+57) (4) 261 95 00 Ext. 9473