

Laboratorio Nro. X

Escribir el tema del laboratorio

Jairo Alonso Carvajal Ochoa
Universidad Eafit
Medellín, Colombia
jacarvajao@eafit.edu.co

Alejandra Toro
Universidad Eafit
Medellín, Colombia
atorog@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Tablas de datos de los algoritmos InsertionSort y MargeSort

InsertionSort			MargeSort	
Tamaño	tiempo(nanoSegundos)		Tamaño	tiempo(nanoSegundos)
5	6400		5	3840
10	39680		10	4267
15	83200		15	4693
20	125440		20	6827
25	200961		25	9814
30	513708		30	6400
35	685228		35	6826
40	502188		40	5973
45	622081		45	8960
50	733868		50	7254
55	780801		55	7254
60	789334		60	4266
65	1169495		65	8533
70	1249282		70	6826
75	1376856		75	7253
80	1276589		80	8960
85	7673187		85	6827
90	2268590		90	4693
95	2747312		95	11093
100	2292057		100	8533

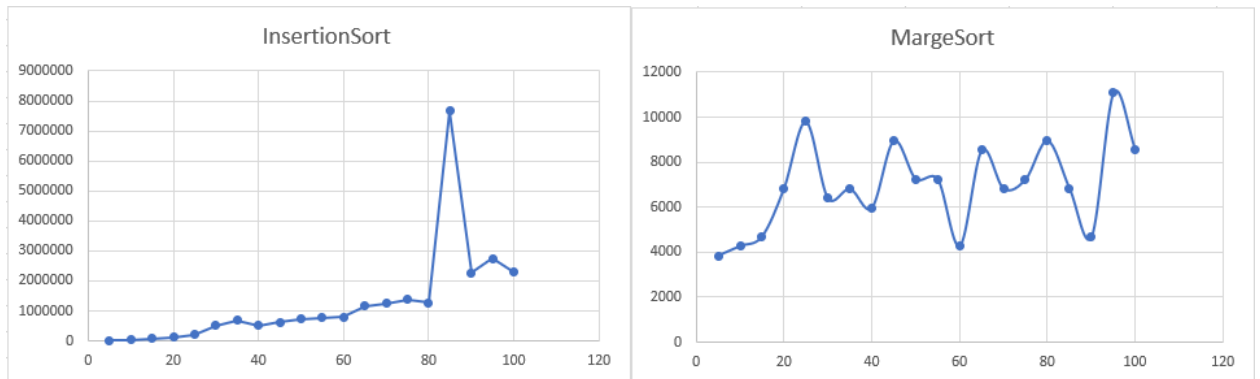
PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473



ESTRUCTURA DE DATOS 1

Código ST0245

3.2 Grafica de Valores de prueba vs tiempo de ejecución



3.3

Complejidad asintótica de InsertionSort:

*El peor caso: $\theta(n^2)$

Complejidad asintótica de MergeSort:

*El peor de los casos: $\theta(n \log n)$

Concluimos que el MergeSort es más eficiente que InsertionSort

3.4.

No es apropiado usar InsertSort para un video juego con millones de datos ya que entre más datos tenga el tiempo va a crecer exponencialmente.

3.5

Basándonos en la plataforma Khan academy, que dice que si todo elemento es mayor o igual a todos los elementos de su izquierda la complejidad es $\theta(n)$. Concluyendo así que el algoritmo en ese caso es más eficiente que MergeSort.

3.6

Inicia con un arreglo, lo que haces que cuando hay dos valores iguales en el arreglo retorna la diferencia de distancia entre i y j.

3.7

<i>Ejercicio</i>	<i>Complejidad</i>
countEvens	$O(n)$
sum13	$O(n)$
lucky13	$O(n)$
fizzArray	$O(n)$
modThree	$O(n)$

3.8

Para todos los ejercicios n es el número de 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

4) Simulacro de Parcial

4.1 No podría ser $O(1)$

4.2 La complejidad asintótica de $mystery(n,m)$ es $O(m \times n)$ ya que en el peor de los casos se va a cumplir el condicional y me va a ingresar al ciclo que va hasta n y mientras más mayor sea n y m la función $p(n)$; se va a hacer poco significativa.

4.3 La complejidad es $O(largo \times ancho)$

4.4 La complejidad asintótica es $O(n^3)$

4.5 La complejidad es $O(n^2)$

4.6 Las instrucciones que ejecuta el algoritmo es: $T(n) = T(n+1) + C$

4.7.

1. El número de pasos que ejecuta en el peor de los casos es: $T(n) = T(n-1)$

2. La complejidad asintótica en el peor de los casos es: $O(n)$

4.9 En el peor de los casos ejecuta más de $n \times m$ pasos

4.11 Ejecuta $T(n) = T(n-1) + T(n-2) + c$ pasos