

Laboratorio Nro. 02

Complejidad de algoritmos

Maria Alejandra Vélez Clavijo
Universidad Eafit
Medellín, Colombia
mavelezc1@eafit.edu.co

Laura Katterine Zapata Rendón
Universidad Eafit
Medellín, Colombia
lkzapatar@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1

Mergesort		Insertsort	
n(tamaño del arreglo)	T(n) (segundos)	n(tamaño del arreglo)	T(n) (segundos)
10000	187	10000	30
20000	295	20000	111
30000	420	30000	222
40000	448	40000	291
50000	658	50000	409
60000	1173	60000	511
70000	1250	70000	769
80000	1484	80000	898
90000	1850	90000	1180
100000	2215	100000	1410
110000	2571	110000	1692
120000	3017	120000	2026
130000	3784	130000	2370
140000	5469	140000	2744
150000	6046	150000	3170
160000	6688	160000	3544
170000	7393	170000	4022
180000	8214	180000	4508
190000	9237	190000	5032
200000	9902	200000	5567

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

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 No sería apropiado utilizar InsertSort para datos muy grandes (millones de datos) puesto que su complejidad es $O(n^2)$ y demoraría bastante tiempo ejecutar el algoritmo con tantos datos, por ello es recomendable que para estos casos se utilice otro algoritmo de ordenación que sea mucho más rápido, como lo podría ser MergeSort.

3.4 La complejidad asintótica logarítmica aparece en el algoritmo MergeSort y esto sucede porque el número de divisiones que ocurre en este algoritmo siempre es menor a n , es decir a medida que aumenta la cantidad de datos, el número de operaciones aumenta, pero no de forma proporcional a los datos.

3.5 En arreglos grandes para que el algoritmo InsertSort sea más rápido que el algoritmo MergeSort, el arreglo debe estar ordenado, que sería el mejor de los casos, ya que la complejidad del InsertSort en este caso en particular es n , mientras que el de MergeSort sigue siendo $n \cdot \log(n)$.

3.6 El ejercicio maxSpan de Array 3 en Coding Bat consiste en un algoritmo que recibe como parámetro un arreglo de enteros y tiene como objetivo retornar un entero el cual es la máxima longitud encontrada. Teniendo como longitud el número de elementos que se encuentran entre dos elementos iguales, incluyéndolos a ellos.

La implementación de este algoritmo consiste en definir dos variables enteras una para almacenar la máxima longitud y otra para almacenar la longitud actual. Luego se ejecuta un ciclo el cual recorre el arreglo desde la primera posición. Dentro de este va otro ciclo el cual recorre el arreglo desde la última posición hasta la primera. Luego se compara si el elemento en la posición de la izquierda es igual al elemento en la posición de la derecha y posteriormente se compara si el número de elementos que hay entre ellos es mayor a la longitud máxima hasta el momento, de ser cierto, la variable donde se almacena la longitud máxima ya pasara a tener el valor de esta longitud.

3.7 y 3.8

Algoritmo	Complejidad para el peor de los casos	n (descripción)
sum13	$O(n)$	cantidad de elementos del arreglo nums
fizzArray	$O(n)$	longitud del arreglo a crear
notAlone	$O(n)$	cantidad de elementos del arreglo nums
linearIn	$O(n)$	longitud de la matriz de enteros exterior
canBalance	$O(n)$	longitud del arreglo nums
countClumps	$O(n)$	longitud del arreglo nums

PhD. Mauricio Toro Bermúdez
 Docente | Escuela de Ingeniería | Info
 Correo: mtorobe@eafit.edu.co | Ofici
 Tel: (+57) (4) 261 95 00 Ext. 9473



ESTRUCTURA DE DATOS 1

Código ST0245

CodingBat	Algoritmo	Descripción	Complejidad para el peor de los casos	n (descripción)
Array 2	sum13	El algoritmo sum13 devuelve la suma de los números de la matriz, devolviendo 0 para una matriz vacía. Excepto que el número 13 no se cuenta y los números que vienen inmediatamente después del 13 tampoco cuentan.	$O(n)$	cantidad de elementos del arreglo nums
	fizzArray	Dado un número n, el algoritmo fizzArray crea y devuelve una nueva matriz int de longitud n, que contiene los números 0, 1, 2, ... n-1	$O(n)$	longitud del arreglo a crear
	notAlone	El algoritmo notAlone devuelve una versión de la matriz dada donde cada instancia del valor dado que está solo se reemplaza por el valor a su izquierda o derecha que sea mayor.	$O(n)$	cantidad de elementos del arreglo nums
	countEvens	El algoritmo countEvens retorna la cantidad de enteros pares en un arreglo.	$O(n)$	longitud del arreglo nums
	lucky13	El algoritmo lucky13 retorna true si no hay 1's o 3's en el arreglo de enteros.	$O(n)$	longitud del arreglo nums
	matchUp	El algoritmo matchUp dados dos arreglos de enteros con la misma longitud, compara cada elemento en la misma posición y determina cuantos de ellos son diferentes y se encuentran en un rango de 2 numeros por encima o 2 numeros por debajo.	$O(n)$	longitud de los dos arreglos nums1 y nums2

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

CodingBat	Algoritmo	Descripción	Complejidad para el peor de los casos	n (descripción)
Array 3	linearIn	El algoritmo linearIn, dadas dos matrices de enteros ordenados en orden creciente, outer e inner , devuelve verdadero si todos los números de la matriz inner aparecen en la matriz outer.	$O(n)$	longitud de la matriz de enteros exterior
	canBalance	El algoritmo canBalance, dada una matriz no vacía, devuelve verdadero si hay un lugar para dividir la matriz de modo que la suma de los números de un lado sea igual a la suma de los números del otro lado.	$O(n)$	longitud del arreglo nums
	countClumps	Supongamos que un "grupo" en una matriz es una serie de 2 o más elementos adyacentes del mismo valor. El algoritmo countClumps devuelve el número de grupos en la matriz dada.	$O(n)$	longitud del arreglo nums
	maxSpan	El algoritmo maxSpan dado un arreglo de enteros, encuentra dos elementos que sean iguales y establece como longitud la cantidad de elementos que hay entre ellos, incluyendolos. Luego devuelve la mayor longitud encontrada.	$O(n^2)$	longitud del arreglo nums
	seriesUp	El algoritmo seriesUp dado un entero n, retorna un arreglo de enteros con la serie de la forma {1, 1, 2, 1, 2, 3, ..., 1, 2, 3 .. n}.	$O(n^2)$	numero enésimo de la serie
	squareUp	El algoritmo squareUp dado un entero n, retorna un arreglo de enteros con la serie de la forma para n=3: {0, 0, 1, 0, 2, 1, 3, 2, 1}.	$O(n^2)$	numero enésimo de la serie

4) Simulacro de Parcial

4.1 Respuesta **C**

4.2 Respuesta **B**

4.3 Respuesta **B**

4.4 Respuesta **A**

4.5

4.5.1 Respuesta **D**

4.5.2 Respuesta **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

ESTRUCTURA DE DATOS 1

Código ST0245

4.6 Para procesar 10000 datos se tardará 100000 segundos

4.7 Respuesta: todas son verdaderas

4.8 Respuesta **A**

4.9 Respuesta **A**

4.10 Respuesta **C**

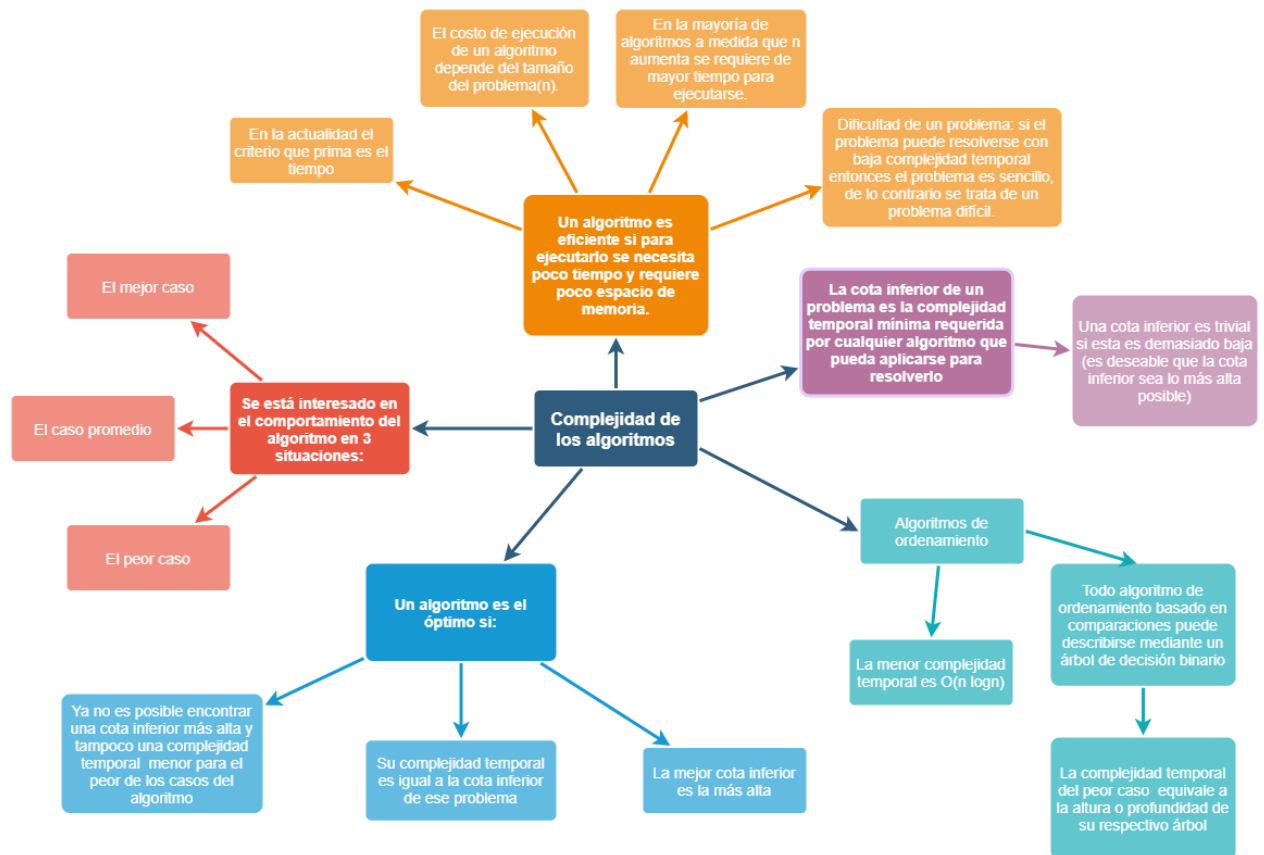
4.11 Respuesta **C**

4.12 Respuesta **B**

4.13 Respuesta **C**

4.14 Respuesta **A**

5) Lectura recomendada (opcional)



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

6) Trabajo en Equipo y Progreso Gradual (Opcional)

6.1 Actas de reunión

Acta	Fecha	Integrante	Hecho
1	31/08/2020	Laura Katterine Zapata Rendón	Leer laboratorio 2
		Maria Alejandra Vélez Clavijo	Leer laboratorio 2
2	2/09/2020	Laura Katterine Zapata Rendón	1.1 Implementar MergeSort e InsertionSort
			3.1 Tomar tiempo de 20 tamaños diferentes para MergeSort y construir tabla
			3.2 Graficar los tiempos de MergeSort
		Maria Alejandra Vélez Clavijo	1.1 Implementar MergeSort e InsertionSort
			3.1 Tomar tiempo de 20 tamaños diferentes para InsertionSort y construir tabla
			3.2 Graficar los tiempos de InsertionSort
3	3/09/2020	Laura Katterine Zapata Rendón	2.1 Resolver 3 ejercicios Array2 CodingBat
			3.5 Complejidad 3 ejercicios Array2 CodinBat
			2.2 Resolver 3 ejercicios Array3 CodingBat
		Maria Alejandra Vélez Clavijo	2.1 Resolver 3 ejercicios Array2 CodingBat
			3.5 Complejidad 3 ejercicios Array2 CodinBat
			2.2 Resolver 3 ejercicios Array3 CodingBa
4	4/09/2020	Laura Katterine Zapata Rendón	3.5 Complejidad 3 ejercicios Array3 CodinBat
			3.3 Responder si es apropiado InsertionSort para grandes cantidades de datos
			3.4 Responder por qué aparece logaritmo en la complejidad del MergeSort
		Maria Alejandra Vélez Clavijo	3.5 Complejidad 3 ejercicios Array3 CodinBat
			3.3 Responder si es apropiado InsertionSort para grandes cantidades de datos
			3.4 Responder por qué aparece logaritmo en la complejidad del MergeSort
5	5/09/2020	Laura Katterine Zapata Rendón	3.4 Explicar cómo funciona maxSpan de Array3 CodingBat
			3.6 Explicar qué son las variables de la complejidad de los ejercicios en línea
			4. Simulacro Parcial
		Maria Alejandra Velez Clavijo	3.4 Explicar cómo funciona maxSpan de Array3 CodingBat
			3.6 Explicar qué son las variables de la complejidad de los ejercicios en línea
			4. Simulacro Parcial

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

Acta	Haciendo	Por hacer
1	1.1 Implementar MergeSort e InsertionSort	3.1 Tomar tiempo de 20 tamaños diferentes para MergeSort y construir tabla
		3.2 Graficar los tiempos de MergeSort
		2.1 Resolver 3 ejercicios Array2 CodingBat
	1.1 Implementar MergeSort e InsertionSort	3.1 Tomar tiempo de 20 tamaños diferentes para InsertionSort y construir tabla
		3.2 Graficar los tiempos de InsertionSort
		2.1 Resolver 3 ejercicios Array2 CodingBat
2	2.1 Resolver 3 ejercicios Array2	3.5 Complejidad 3 ejercicios Array2 CodinBat
		2.2 Resolver 3 ejercicios Array3 CodingBat
		3.5 Complejidad 3 ejercicios Array3 CodinBat
	2.1 Resolver 3 ejercicios Array2	3.5 Complejidad 3 ejercicios Array2 CodinBat
		2.2 Resolver 3 ejercicios Array3 CodingBat
		3.5 Complejidad 3 ejercicios Array3 CodinBat
3	3.5 Complejidad 3 ejercicios Array3 CodinBat	3.3 Responder si es apropiado InsertionSort para grandes cantidades de datos
		3.4 Responder por qué aparece logaritmo en la complejidad del MergeSort
		3.5 Responder cuando InsertionSort es más rapido que MergeSort
	3.5 Complejidad 3 ejercicios Array3 CodinBat	3.3 Responder si es apropiado InsertionSort para grandes cantidades de datos
		3.4 Responder por qué aparece logaritmo en la complejidad del MergeSort
		3.5 Responder cuando InsertionSort es más rapido que MergeSort
4	3.5 Responder cuando InsertionSort es más rapido que MergeSort	3.4 Explicar cómo funciona maxSpan de Array3 CodingBat
	4. Simulacro Parcial	3. 6 Explicar qué son las variables de la complejidad de los ejercicios en linea
	3.5 Responder cuando InsertionSort es más rapido que MergeSort	3.4 Explicar cómo funciona maxSpan de Array3 CodingBat
	4. Simulacro Parcial	3. 6 Explicar qué son las variables de la complejidad de los ejercicios en linea
5	5 Lectura recomendada y realización mapa	
	5 Lectura recomendada y realización mapa	

6.2 El reporte de cambios en el código

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

[kattezapata](#) / [ST0245-001](#)

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

History for [ST0245-001](#) / [laboratorios](#) / [lab02](#)

Commits on Sep 5, 2020

Se organiza la complejidad de MergeSort  kattezapata committed 1 hour ago
Código para hallar los 20 tiempos de insertionSort  kattezapata committed 2 hours ago
Se implementa ejercicio squareUp de Array3 CodingBat  kattezapata committed 2 hours ago
Se implementa ejercicio seriesUp de Array3 CodingBat  kattezapata committed 2 hours ago
Se implementa ejercicio maxSpan de Array3 CodingBat  kattezapata committed 3 hours ago
Se implementa ejercicio matchUp de Array2 CodingBat  kattezapata committed 3 hours ago
Se implementa ejercicio lucky13 de Array2 CodingBat  kattezapata committed 4 hours ago
Se implementa ejercicio countEvens de Array2 CodingBat  kattezapata committed 4 hours ago
Título Array2 y Array3 


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



 kattezapata committed 5 hours ago

-> Commits on Sep 3, 2020


No commit message

 malejav02 committed 2 days ago

Se añaden 3 ejercicios de Array3(Codingbat) con su documentación.

 malejav02 committed 2 days ago


-> Commits on Sep 2, 2020


Se añaden tres ejercicios de codingbat (Array2) con su respectiva do...

 malejav02 committed 3 days ago


-> Commits on Sep 1, 2020

Implementación algoritmo mergesort y documentación del mismo

 malejav02 committed 4 days ago


-> Commits on Aug 31, 2020

Add files via upload

 kattezapata committed 5 days ago

Delete Código.java

 kattezapata committed 5 days ago

Add files via upload

 kattezapata committed 5 days ago

-> Commits on Aug 1, 2017

Agregando estructura

 jarcil13 committed on 1 Aug 2017

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

ESTRUCTURA DE DATOS 1

Código ST0245

Word LAB02 - Guardado

Volver al documento Ahora mismo Guardar una copia

ESTRUCTURA DE DATOS 1
Código ST0245

Laboratorio Nro. 02
Complejidad de algoritmos

Maria Alejandra Vélez Clavijo
Universidad Eafit
Medellín, Colombia
mavelezc1@eafit.edu.co

Laura Katterine Zapata Rendón
Universidad Eafit
Medellín, Colombia
lkzapatar@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1

Mergesort		Insertsort	
n (tamaño del arreglo)	T(n) (segundos)	n (tamaño del arreglo)	T(n) (segundos)
10000	30		

Historial de versiones

Mostrar cambios

Hoy

Ahora mismo

- Laura Katterine Zapata Rendón

hace 2 horas

- Laura Katterine Zapata Rendón

hace 2 horas

- Laura Katterine Zapata Rendón

hace 2 horas

- Laura Katterine Zapata Rendón

hace 2 horas

- Laura Katterine Zapata Rendón

hace 3 horas

- Laura Katterine Zapata Rendón

hace 6 horas

- Maria Alejandra Vélez Clavijo
- Laura Katterine Zapata Rendón

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

Historial de versiones
Mostrar cambios ☒

Hoy

Ahora mismo

L Laura Katterine Zapata Rendón

hace 2 horas

L Laura Katterine Zapata Rendón

hace 2 horas

L Laura Katterine Zapata Rendón

hace 2 horas

L Laura Katterine Zapata Rendón

hace 2 horas

L Laura Katterine Zapata Rendón

hace 3 horas

L Laura Katterine Zapata Rendón

hace 6 horas

M Maria Alejandra Vélez Clavijo

L Laura Katterine Zapata Rendón

Historial de versiones
Mostrar cambios ☒

hace 6 horas

M Maria Alejandra Vélez Clavijo

L Laura Katterine Zapata Rendón

hace 7 horas

M Maria Alejandra Vélez Clavijo

hace 7 horas

M Maria Alejandra Vélez Clavijo

hace 7 horas

M Maria Alejandra Vélez Clavijo

L Laura Katterine Zapata Rendón

hace 7 horas

M Maria Alejandra Vélez Clavijo

L Laura Katterine Zapata Rendón

hace 7 horas

L Laura Katterine Zapata Rendón

hace 7 horas

M Maria Alejandra Vélez Clavijo

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

Historial de versiones

Mostrar cambios ☒

- hace 7 horas
M Maria Alejandra Vélez Clavijo
- hace 7 horas
M Maria Alejandra Vélez Clavijo
L Laura Katterine Zapata Rendón
- hace 7 horas
M Maria Alejandra Vélez Clavijo
L Laura Katterine Zapata Rendón
- hace 7 horas
L Laura Katterine Zapata Rendón
- hace 7 horas
M Maria Alejandra Vélez Clavijo
- hace 7 horas
M Maria Alejandra Vélez Clavijo
- hace 7 horas
M Maria Alejandra Vélez Clavijo

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