

	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST245
		Estructura de Datos 1

Laboratorio Nro. 2: Notación O grande.

Jose Joab Romero Humba
Universidad Eafit
Medellín, Colombia
jjromeroh@eafit.edu.co

Kevin Alexander Herrera Garcés
Universidad Eafit
Medellín, Colombia
kaherrerag@eafit.edu.co

Manuel Alejandro Gutiérrez Mejía
Universidad Eafit
Medellín, Colombia
magutierrm@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

1.

Insertion sort:

```
run:
Tamaño del arreglo 5 Tiempo: 19
Tamaño del arreglo 25 Tiempo: 386
Tamaño del arreglo 45 Tiempo: 1218
Tamaño del arreglo 65 Tiempo: 2622
Tamaño del arreglo 85 Tiempo: 4999
Tamaño del arreglo 105 Tiempo: 7245
Tamaño del arreglo 125 Tiempo: 10981
Tamaño del arreglo 145 Tiempo: 13110
Tamaño del arreglo 165 Tiempo: 18660
Tamaño del arreglo 185 Tiempo: 23996
Tamaño del arreglo 205 Tiempo: 29578
```

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

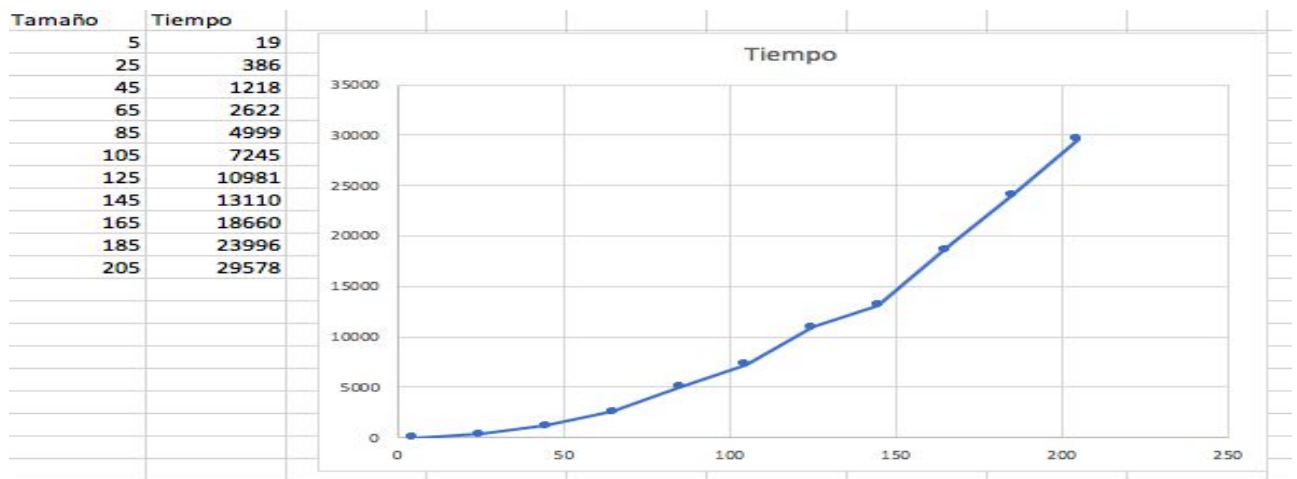
	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST245
		Estructura de Datos 1

Merge Sort:

```
run:
Tamaño del arreglo 5 Tiempo: 10
Tamaño del arreglo 25 Tiempo: 58
Tamaño del arreglo 45 Tiempo: 107
Tamaño del arreglo 65 Tiempo: 164
Tamaño del arreglo 85 Tiempo: 208
Tamaño del arreglo 105 Tiempo: 247
Tamaño del arreglo 125 Tiempo: 296
Tamaño del arreglo 145 Tiempo: 350
Tamaño del arreglo 165 Tiempo: 406
Tamaño del arreglo 185 Tiempo: 493
Tamaño del arreglo 205 Tiempo: 547
```

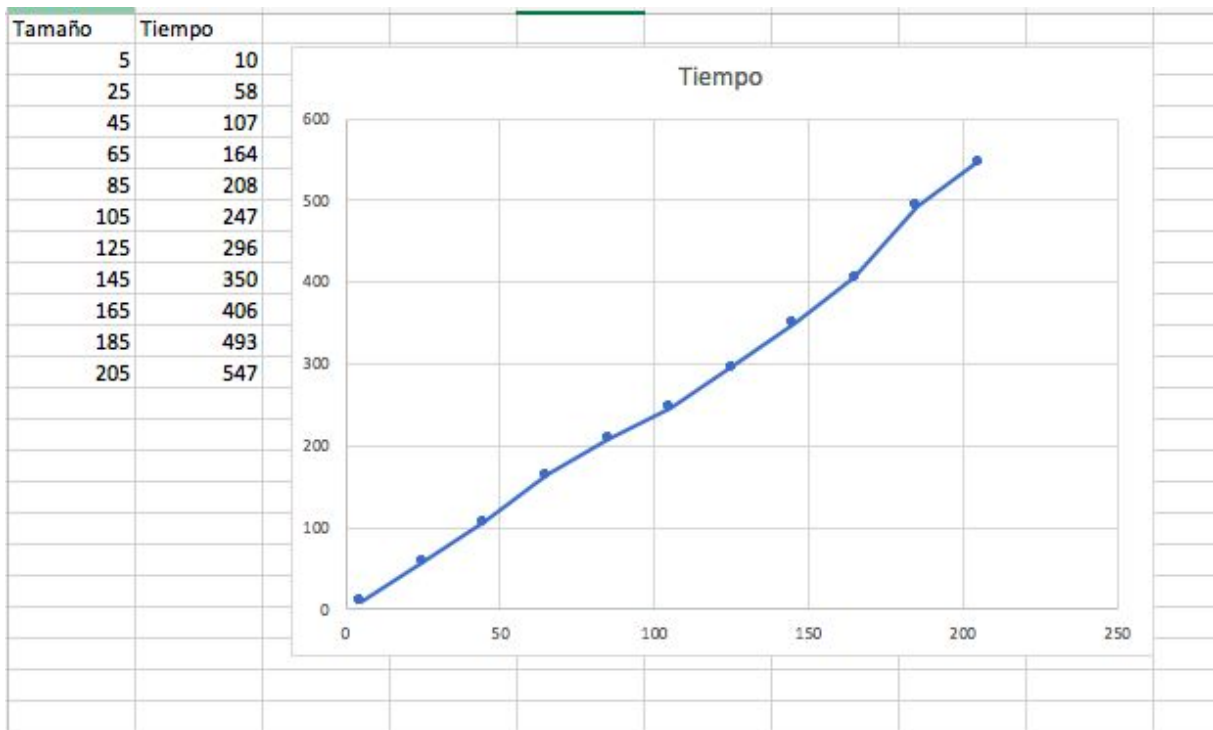
2.

Insertion sort:



	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST245
		Estructura de Datos 1

Merge Sort:



3. El algoritmo merge sort aunque requiere más código que insertion sort, es más eficiente para arreglos más grandes ya que divide el problema en dos mitades de un tamaño similar para así sustraer el problema de una manera más sencilla y poder hacer las sustituciones correspondientes, el problema utiliza la filosofía de divide y vencerás, en cambio insertion sort al solo ir de paso en paso será más complicado para arreglos de millones de elementos.
4. La solución del algoritmo maxSpan de coding bat funciona de la siguiente manera, debemos recorrer el arreglo y crear una variable auxiliar la cual inicializará en elementos menos uno después debemos crear un ciclo while el cual nos permitirá tener control de nuestro arreglo con nuestra variable auxiliar e ir disminuyendo, aquí se realizará un cálculo el cual será restarle a la variable auxiliar nuestro iterador + 1, por último se comparará el caso y se retornará el mayor intervalo.

4) Simulacro de Parcial

1. C
2. D
3. B
4. B

	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST245
		Estructura de Datos 1

5. *D*
6. *A*
7. $1 t(n-1)+c$
8. *B*
9. *D*
10. *C*
11. *C*
12. *B*
13. *A*

5. Cálculo de Complejidad:

- CountEvents: $T(n) = O(n)$
- bigDiff: $T(n) = O(m)$
- Sum13: $T(n) = O(n)$
- has22: $T(n) = O(n)$
- lucky13: $T(n) = O(n)$
- canBalance: $T(n) = O(n)$
- countClumps: $T(n) = O(n*m)$
- fix45: $T(n) = O(n*m)$
- seriesUp: $T(n) = O(n*m)$
- fix34: $T(n) = O(n*m)$

6. explicacion de variables

- CountEvents:
variable n: es la cantidad de elementos que hay en el arreglo, esto debido a que el programa los tiene que recorrer todos
- bigDiff:
variable m: es la longitud del arreglo debido a que los tiene que recorrer todos para poder conocer cual es el mayor y el menor
- Sum13:
variable n: es la longitud del arreglo, debido a que se tiene que recorrer todas las posiciones para poder hacer la sumatoria de las mismas
- has22:
variable n: es la cantidad de elementos en el arreglo, siendo esto en el peor de los casos.
- lucky13:
variable n: es la longitud del arreglo, siendo esto en el peor de los casos
- canBalance:
variable n: en el peor de los casos el programa tiene que recorrer todo el arreglo, siendo n la longitud del mismo
- countClumps:
variable n: es la cantidad de elementos en el arreglo que no se repiten
variable m: es la cantidad de veces que un elemento se repite en el arreglo

	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST245
		Estructura de Datos 1

- **fix45:** las variables n y m hacen referencia a los ciclos presentes en el código para mantener el control en el arreglo y organizar el nuevo arreglo con las condiciones especificadas
- **seriesUp**
variable n: es la longitud del arreglo
variable m: es la cantidad de veces que se visitan los elementos del arreglo para sumarle el valor de j y así poder dar forma al patrón
- **fin34:** las letras n y m hacen referencia a los ciclos que se encuentran dentro del código, se encuentran estrictamente relacionadas a la cantidad de elementos presentes en el arreglo y son estos quienes permiten llevar el control en el ciclo