ESTRUCTURA DE DATOS 1
Código ST0245

Laboratorio Nro. 2
Complejidad de algoritmos

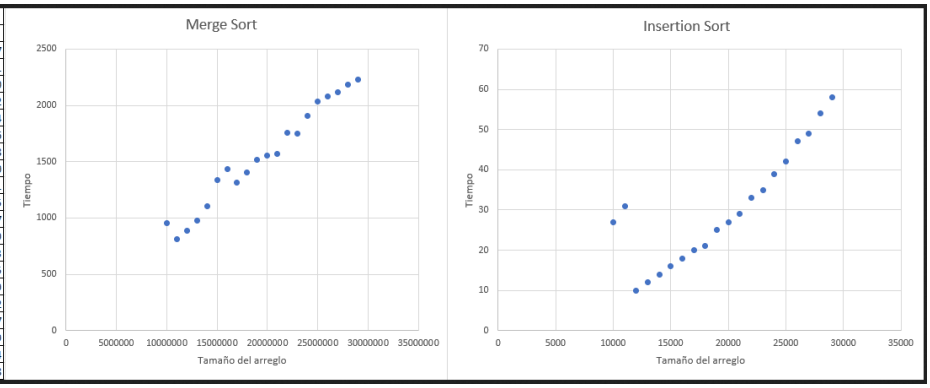
Esteban Bernal Correa
Universidad Eafit
Medellín, Colombia
ebernalc@eafit.edu.co

Juan Manuel Garzón
Universidad Eafit
Medellín, Colombia
jmgarzonv@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 y 3.2

Merge Sort		Insertion Sort	
Tamaño	Tiempo(ms)	Tamaño	Tiempo(ms)
10000000	954	10000	27
11000000	815	11000	31
12000000	884	12000	10
13000000	976	13000	12
14000000	1105	14000	14
15000000	1335	15000	16
16000000	1436	16000	18
17000000	1317	17000	20
18000000	1408	18000	21
19000000	1519	19000	25
20000000	1554	20000	27
21000000	1570	21000	29
22000000	1756	22000	33
23000000	1750	23000	35
24000000	1905	24000	39
25000000	2032	25000	42
26000000	2081	26000	47
27000000	2118	27000	49
28000000	2184	28000	54
29000000	2227	29000	58



3.3 Es recomendable a comparación del mage sort, ya que el procesamiento datos se haría en muy pocos segundos, por lo que puede sr usado en videojuegos, pero una complejidad así puede generar mucho mas delay a la hora dejugar en línea

3.4 porque el método mergeSort tiene log, porque tiene 2 métodos, uno divide el arreglo en n veces, y el otro lo organiza.

3.4 (Opcional) El ejercicio maxSpan funciona de de esta manera.
Primero se inicializan 2 variables internas "span" y "aux" donde se ira guardando nuestro progreso, luego en el primer for se recorre todo el arreglo para comparar el primer número el segundo for nos da el segundo número para comparar.
En base a como esta organizado, lo que se busca hacer es que se compare el primer elemento del arreglo con todos los demás elementos del arreglo, luego se intenta con el segundo, y de esta forma se va comparando cada elemento del arreglo con los demás.

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

Después de esto llegamos al if el cual se basa en la condición de igualdad entre nuestros 2 números a comparar:
Si son iguales, en aux se calcula el valor absoluto de i-j(si se restan nos dan su distancia entre las 2). El +1 es de gran utilidad ya que los arreglos utilizan por su definición la posición 0 como inicial, después de esto, si la diferencia de posiciones es mayor que alguna diferencia de posición anterior, si es cierto, este se convertirá en nuestro nuevo "span", pero si es falso, se sigue comparando con el siguiente elemento;
Ya con la ayuda de los ciclos todo este proceso se va a repetir hasta comparar todos los elementos del arreglo.

3.5 De los ejercicios del 2.1 resueltos todos tienen complejidad O(n).
De los ejercicios de la sección 2.2:

Fix34:O(n^2)
CanBalance:O(n)
MaxSpanO(n^2)
LinearIn:O(n)
SquareUp:O(n^2)

3.5 (Opcional): Para arreglos grandes si los datos están en orden o todos son iguales el algoritmo de insertion sort va a ser mucho mas rápido que el marge sort

3.6 n es el tamaño del arreglo

4) Simulacro de Parcial

4.1 100ms
4.2 D
4.3 A
4.4 .
***4.5* A**
***4.6* A**
***4.7* A,B,C**

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