Laboratorio Nro. 2: Notación O grande

Santiago Pulgarin Vasquez
Universidad Eafit
Medellín, Colombia
spulgarinv@eafit.edu.co

Juan Camilo Guerrero Alarcón
Universidad Eafit
Medellín, Colombia
jcguerrera@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1)

Tablas de los Algoritmos :

Insertion Sort

Tamaño	Tiempo (ms)
10000	19
20000	71
30000	175
40000	300
50000	491
60000	615
70000	842
80000	1114
90000	1431
100000	1776
110000	2158
120000	2609
130000	3053
140000	3560
150000	4101
160000	4668
170000	5319
180000	5988
190000	6660
200000	7371

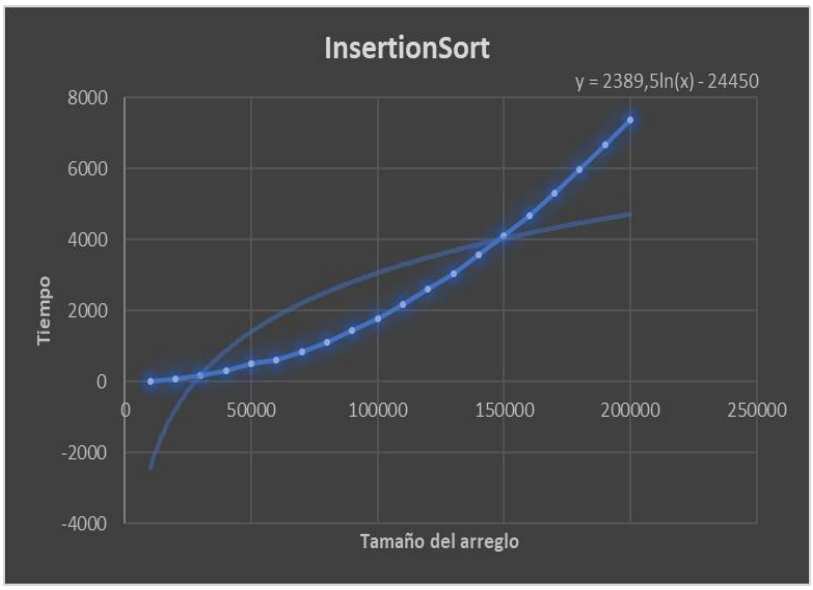
Marge Sort

Tamaño	Tiempo (ms)
100000	49
200000	68
300000	45
400000	51
500000	107
600000	85
700000	155
800000	104
900000	229
1000000	153
1100000	129
1200000	153
1300000	303
1400000	170
1500000	211
1600000	313
1700000	392
1800000	305
1900000	276
2000000	342

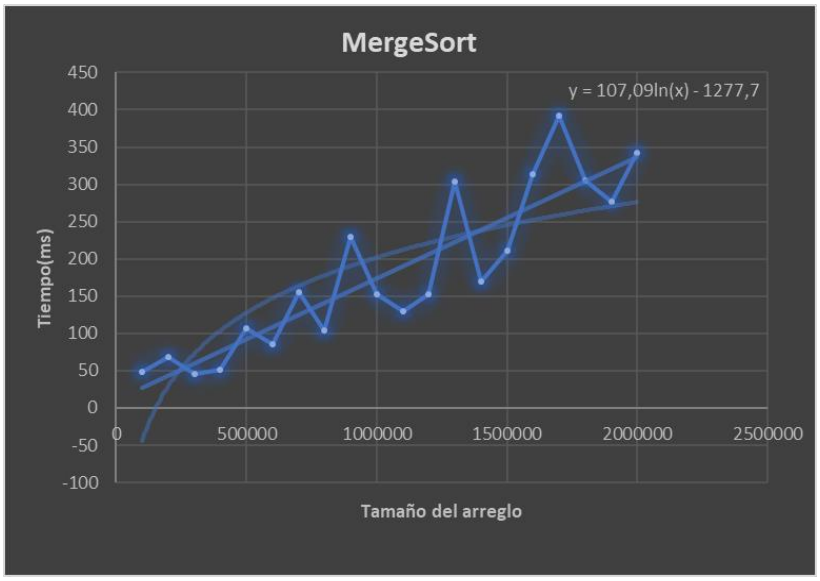
3.2)

Graficas:

Insertion Sort.



Merge Sort.



3.3)

El algoritmo denominado como Merge Sort con respecto al Insertion Sort es mucho más efectivo ya que a una gran cantidad de datos su tiempo de ejecución es mucho más optimo y eso lo hace mucho más eficiente con respecto al otro algoritmo, el Insertion es más conocido pero esta plateado para una serie de datos mucho menos ya que su complejidad asintótica es de O(n^2) lo que no es tan favorable, sin embargo no lo hace menos efectivo pero si menos optimo con respecto al Merge, en conclusión el algoritmo de ordenamiento para manejar bases de datos de grandes valores y que sea mucho más rápido es sin duda el Merge Sort.

3.4)

El ejercicio denominado como MaxSpan básicamente funciona por medio de dos ciclos, uno simplemente se encarga de recorrer todos los elementos que tengamos en nuestro array y el segundo ciclo lo que va a hacer es buscar un elemento que se repita en dicho arreglo, para así determinar un lapso o una cadena y así va a verificar cual es la mayor cantidad de elementos que se encuentran entre determinado lapso y lo que va a proceder es a sumarlos y retornar dicha suma del lapso mayor, en caso de que no encuentre un lapso entre dos numero la implementación de este algoritmo va a retornar el span o lapso más largo de este arreglo.

3.5)

Array2

countEvens

T(n) = c1 + c2 + c3(n+1) + c4n + c5
T(n) es O (c1 + c2 + c3(n+1) + c4n + c5) Por definición de O.
T(n) es O (c3(n+1)) Por regla de la suma.
T(n) es O (n+1) Por regla del producto.
T(n) es O (n) Por regla de la suma.

has22

T(n) = c1 + c2(n+1) + c3n + c4 + c5
T(n) es O (c1 + c2(n+1) + c3n + c4 + c5) Por definición de O.
T(n) es O (c2(n+1)) Por regla de la suma.
T(n) es O (n+1) Por regla del producto.
T(n) es O (n) Por regla de la suma.

bigDiff

T(n) = c1 + c2 + c3 + c4(n+1) + c5n + c6 + c7n + c8 + c9
T(n) es O (c1 + c2 + c3 + c4(n+1) + c5n + c6 + c7n + c8 + c9) Por definición de O.
T(n) es O (c4(n+1)) Por regla de la suma.
T(n) es O (n+1) Por regla del producto.
T(n) es O (n) Por regla de la suma.

sum13

T(n) = c1 + c2 + c3(n+1) + c4n + c5 + c6
T(n) es O (c1 + c2 + c3(n+1) + c4n + c5 + c6) Por definición de O.

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

T(n) es O (c3(n+1)) Por regla de la suma.
T(n) es O (n+1) Por regla del producto.
T(n) es O (n) Por regla de la suma.

lucky13

T(n) = c1 + c2(n+1) + c3n + c4 + c5
T(n) es O (c1 + c2(n+1) + c3n + c4 + c5) Por definición de O.
T(n) es O (c2(n+1)) Por regla de la suma.
T(n) es O (n+1) Por regla del producto.
T(n) es O (n) Por regla de la suma.

Array3

maxSpan

T(n) = c1 + c2 + c3 + c4(n+1) + c5 + c6(n+1)n + c7 + c8(n+1)n + c9 + c10 + c11 + c12 + c13 + c14
T(n) es O (c1 + c2 + c3 + c4(n+1) + c5 + c6(n^2+n) + c7 + c8(n^2+n) + c9 + c10 + c11 + c12 + c13 + c14) Por definición de O.
T(n) es O (c6(n^2+n)) Por regla de la suma.
T(n) es O (n^2+n) Por regla del producto.
T(n) es O (n^2) Por regla de la suma.

fix34

T(n) = c1 + c2 + c3(n+1) + c4n + c5 + c6(n+1)n + c7 + c8(n+1)n + c9 + c10 + c11 + c12
T(n) es O (c1 + c2 + c3(n+1) + c4n + c5 + c6(n^2+n) + c7 + c8(n^2+n) + c9 + c10 + c11 + c12) Por definición de O.
T(n) es O (c6(n^2+n)) Por regla de la suma.
T(n) es O (n^2+n) Por regla del producto.
T(n) es O (n^2) Por regla de la suma.

fix45

T(n) = c1 + c2(n+1) + c3n + c4n + c5 + c6(n+1)n + c7 + c8(n+1)n + c9 + c10 + c11 + c12 + c13
T(n) es O (c1 + c2(n+1) + c3n + c4n + c5 + c6(n^2+n) + c7 + c8(n^2+n) + c9 + c10 + c11 + c12 + c13) Por definición de O.
T(n) es O (c6(n^2+n)) Por regla de la suma.
T(n) es O (n^2+n) Por regla del producto.
T(n) es O (n^2) Por regla de la suma.

canBalance

T(n) = c1 + c2(n+1) + c3n + c4 + c5(n+1)n + c6n + c7 + c8(n+1)n + c9n + c10 + c11 + c12
T(n) es O (c1 + c2(n+1) + c3n + c4 + c5(n^2+n) + c6n + c7 + c8(n^2+n) + c9n + c10 + c11 + c12) Por definición de O.
T(n) es O (c5(n^2+n)) Por regla de la suma.
T(n) es O (n^2+n) Por regla del producto.
T(n) es O (n^2) Por regla de la suma.

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

squareUp

T(n) = c1 + c2 + c3 + c4(n+1) + c5 + c6(n+1)n + c7n + c8 + c9(n+1)n + c10n + c11
T(n) es O (c1 + c2 + c3 + c4(n+1) + c5 + c6(n^2+n) + c7n + c8 + c9(n^2+n) + c10n + c11) Por definición de O.
T(n) es O (c6(n^2+n)) Por regla de la suma.
T(n) es O (n^2+n) Por regla del producto.
T(n) es O (n^2) Por regla de la suma.

3.6)

En el calculo de la complejidad en el numeral anterior, la variable "n" significa el tamaño del problema, entonces dependiendo de esto, el problema se expande n+1, o en los ciclos, cuando se da la circunstancia de los ciclos anidados, el ciclo mas interno se repite n veces.

4) Simulacro de Parcial

1. O(n+m).
2. O(m*n).
3. O(ancho).
4. O(n^3).
5. O(n^2).
6. T(n) = T(n-1) + C.
7. T(n)=(n-1) O(n).
8. O(n*sqrt(n)).
9. Ejecuta más de (n x m) pasos.
10. Ejecuta menos de (n x logn) pasos.
11. Ejecuta T(n) = T(n-1) + T(n-1) + C
12. O(m*sqrt(n)).
13. O(n^3).

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