Laboratorio Nro. 2: Notación O grande

Isabella Arango Restrepo
Universidad Eafit
Medellín, Colombia
iarangor1@eafit.edu.co

Juan David Rengifo Castro.
Universidad Eafit
Medellín, Colombia
jdrengifoc@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

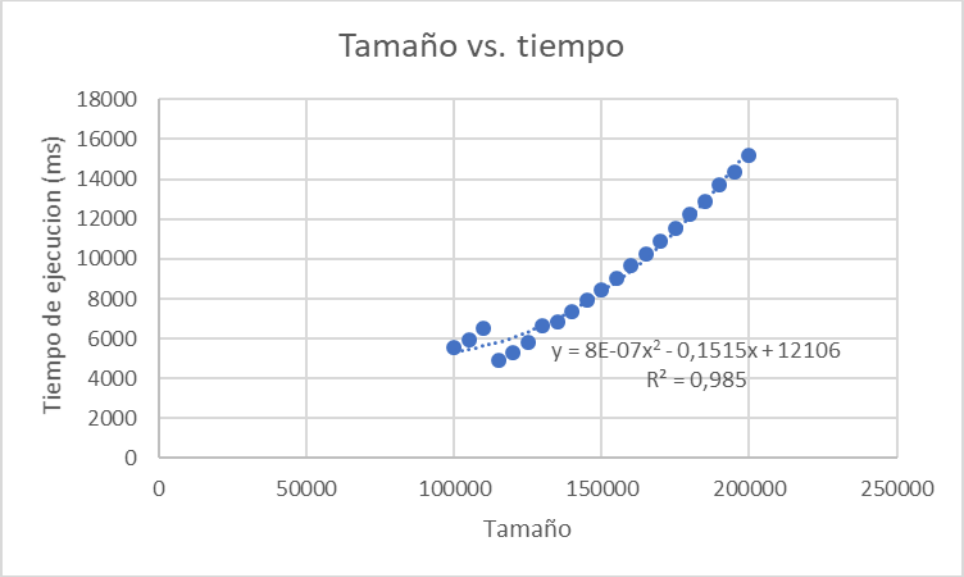
1. Insertion Sort

Tamaño	Tiempo de ejecución(ms)
100000	5583
105000	5942
110000	6528
115000	4882
120000	5327
125000	5781
130000	6618
135000	6822
140000	7346
145000	7920
150000	8437
155000	9048
160000	9656
165000	10275
170000	10889
175000	11557
180000	12209
185000	12892
190000	13713
195000	14387
200000	15216

Mergesort

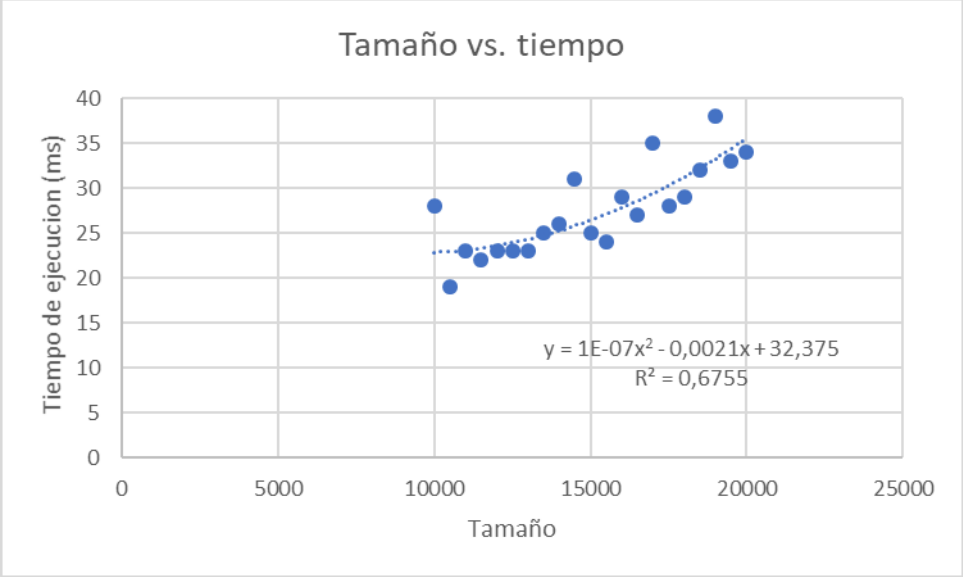
Tamaño del arreglo	Tiempo de ejecución (ms)
100000	28
105000	19
110000	23
115000	22
120000	23
125000	23
130000	23
135000	25
140000	26
145000	31
150000	25
155000	24
160000	29
165000	27
170000	35
175000	28
180000	29
185000	32
190000	38
195000	33
200000	34

2. Gráfica para los tiempos de Insertion Sort



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

Gráfica de tiempos para Mergesort



3. Teniendo en cuenta las gráficas anteriores y considerando que se tomaron tiempos para arreglos de igual tamaño con InsertionSort y con Mergesort, se puede concluir que este último toma menos tiempo en ordenar arreglos que varían el tamaño entre 100.000 y 200.000. Por lo tanto, podríamos concluir que el mergesort es mucho más eficiente al momento de ordenar arreglos de grandes tamaños y por esta razón no es adecuado usar el método de InsertionSort en bases de datos con millones de elementos, pues tardaría demasiado tiempo en alcanzar el objetivo.
4. El ejercicio maxSpan de Coding Bat de Array3 busca encontrar la cantidad máxima de elementos que hay entre un número que aparece mínimo dos veces en el arreglo. Para esto, revisa primero si el arreglo tiene un tamaño mayor que cero, pues de lo contrario retornará que hay cero elementos entre dos números. En caso de que sea mayor a cero, entonces establece un entero a=1 que indica el número mínimo de elementos que hay entre un valor y otro o el mismo. Además, estable un contador que representará la cantidad de elementos entre dos valores del arreglo que representan el mismo número, pero están en diferente posición. Finalmente, si el contador llega a exceder al entero a ya establecido, entonces este tomará el valor del contador y se retornará.
5. Array2
    a. lucky13: T(n) = cn + c
    b. has77: T(n) = c(n-1) + c(n-2) + c
    c. shiftLeft: T(n) = c(n-1) + c
    d. Pre4: T(n,m) = cn + cm + c
    e. fizzBuzz: T(n,m) = c(m-n) + c

    Array3
    f. maxSpan: T(n) = c(n-1)(n-1) + c(n-1) + c
    g. Fix34: T(n) = cnn + cn + c
    h. Fix45: T(n) = cnn + cn + c
    i. linearIn: T(n,n) = cmn + cm + c
    j. canBalance: T(n) = c(n-1) + c(n-2) + c
6. Array2
    a. lucky13: n: longitud de nums

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

b. has77: T(n) = n: longitud de nums
c. shiftLeft: n: longitude de nums
d. Pre4: T(n,m) = n: longitud de nums, m: longitud de a
e. fizzBuzz: n: start, m: end

Array3

f. maxSpan: n: longitude de nums
g. Fix34: n: longitud de nums
h. Fix45: n: longitud de nums
i. linearIn: n: longitud de outer, m: longitud de inner
j. canBalance: n: longitud de nums

4) Simulacro de Parcial

1. *c*
2. *d*
3. *b*
4. *b*
5. d
6. a
7. T(n)=T(n-1) + C, O(n)
8. b
9. d
10. c
11. c
12. b
13. a

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