

Taller 5

Complejidad y gráficas del tiempo de ejecución de los algoritmos propuestos

Valentina Moreno Ramírez

Universidad Eafit

Medellín, Colombia

vmorenor@eafit.edu.co

Alejandra Palacio Jaramillo

Universidad Eafit

Medellín, Colombia

apalacioj@eafit.edu.co

Ejercicio 1:

n	T(n)
10000	24
11000	19
12000	20
13000	19
14000	21
15000	21
16000	22
17000	24
18000	28
19000	29
20000	33
21000	37
22000	40
23000	45
24000	47
25000	52
26000	57
27000	60
28000	67
29000	69
30000	75
31000	80
32000	85
33000	89
34000	96
35000	101
36000	108
37000	113



Complejidad asintótica para el peor de los casos del algoritmo *insertion sort*:

$$T(n) = c_1 + c_2 * n - c_2 + c_3 * n - c_3 + c_4 + c_5 * n^2 - 2c_5 * n + c_5 + c_6 * n^2 - 2c_6 * n + c_6 + c_7 * n + c_7$$

$T(n) = O(c_1 + c_2 * n - c_2 + c_3 * n - c_3 + c_4 + c_5 * n^2 - 2c_5 * n + c_5 + c_6 * n^2 - 2c_6 * n + c_6 + c_7 * n + c_7)$,
aplicando regla de reflexividad.

$T(n) = O(c_6 * n^2)$, aplicando regla de la suma.

$T(n) = O(n^2)$, aplicando regla del producto

Entonces, la complejidad asintótica para el peor de los casos de este algoritmo es $O(n^2)$, es decir, cuadrática, donde n es igual a la longitud del arreglo.

Teniendo en cuenta este planteamiento, es correcto afirmar que, aunque el insertion sort tardó 0,03 segundos en ordenar un arreglo de 155000 elementos, no es apropiado para ordenar grandes volúmenes de datos porque al ser la complejidad cuadrática, significa que entre más datos haya, mucho mayor va a ser el tiempo de ejecución, en otras palabras, entre más grande sea n , el tiempo de ejecución será de n al cuadrado. Además, hay que tener en cuenta qué tan ordenado estaba el arreglo antes de aplicar el insertion sort ya que no será la mismo ordenar un arreglo con todos sus elementos en desorden que un arreglo solo con unos cuantos elementos en desorden.

Ejercicio 2:

n	t(n)
1000000	5
2000000	12
3000000	13
4000000	14
5000000	17
6000000	19
7000000	73
8000000	60
9000000	33
10000000	23
11000000	132
12000000	23
13000000	61
14000000	30
15000000	49
16000000	39
17000000	30
18000000	71
19000000	28
20000000	38
21000000	81
22000000	80
23000000	33
24000000	87
25000000	32
26000000	35
27000000	34
28000000	35



Complejidad asintótica para el peor de los casos del algoritmo del *punto 2*:

$$T(n) = c_1 + c_2 + c_3 * n + c_4 * n + c_5 * n$$

$$T(n) = O(c_1 + c_2 + c_3 * n + c_4 * n + c_5 * n), \text{ aplicando regla de reflexividad.}$$

$$T(n) = O(c_5 * n), \text{ aplicando regla de la suma.}$$

$$T(n) = O(n), \text{ aplicando regla del producto.}$$

Entonces, la complejidad asintótica para el peor de los casos del algoritmo que suma los elementos de un arreglo utilizando ciclos es $O(n)$, es decir, lineal, donde n es igual a la longitud del arreglo.

De acuerdo con este ejercicio, se puede afirmar que existe una gran diferencia del tiempo de ejecución en la implementación con ciclos y la implementación recursiva de un algoritmo, lo cual es debido al funcionamiento de cada uno de estos métodos. Por ejemplo, cuando un algoritmo es recursivo, se llama así mismo las veces que se requieran, por tanto, es un proceso más complejo que la iteración, lo cual significa que tendrá una complejidad mayor en la mayoría de los casos.