

PRACTICA 1.2 –

EJERCICIO 1 –

n	TBucle1	TBucle2	TBucle3	TBucle4
100	0	0	1	0
200	0	1	4	6
400	0	3	15	43
800	0	13	64	324
1600	0	51	263	2567
3200	0	228	1100	20506
6400	1	935	4658	FDT
12800	2	4126	19796	FDT
25600	3	18480	FDT	FDT
51200	6	FDT	FDT	FDT

Razone si los diferentes tiempos obtenidos concuerdan con lo esperado, según la complejidad temporal de los cuatro casos.

Las complejidades de tiempo encajan con los tiempos mostrados por los bucles. Es importante notar la diferencia de tiempo entre el Bucle2 y Bucle3.

Ambos tienen complejidad $O(n^2 * \log n)$, sin embargo, tienen tiempos diferentes. ¿Por qué?

Tienen tiempos diferentes debido a que mientras que en el bucle 2, el bucle se ejecuta un número logarítmico de veces con el bucle interno siendo de complejidad n^2 , el bucle 3 se ejecuta un número n^2 de veces con el bucle interno siendo de complejidad logarítmica.

EJERCICIO 2 –

Implementar tres nuevas clases Bucle5, Bucle6 y Bucle7, que simulen algoritmos iterativos con una complejidad $O(n^2 \log 2 n)$, $O(n^3 \log n)$ y $O(n^4)$ respectivamente.

Tras implementar las clases, mida sus tiempos de ejecución y rellene la tabla:

n	TBucle5	TBucle6	TBucle7
100	5	65	20
200	24	539	284
400	115	4594	4364
800	535	39410	FDT
1600	2562	FDT	FDT
3200	11711	FDT	FDT
6400	53367	FDT	FDT
12800	FDT	FDT	FDT
25600	FDT	FDT	FDT
51200	FDT	FDT	FDT

Razone si los diferentes tiempos obtenidos concuerdan con lo esperado, según la complejidad temporal de los tres casos.

Encajan según sus complejidades la curva que describen, el bucle 7, por supuesto, es el primero en quedarse fuera de tiempo (debido a que su complejidad es la más alta) siendo seguido por el Bucle6 (que tiene complejidad n^3)

EJERCICIO 3 –

n	TBucle1	TBucle2	T1/T2
100	1	1	1
200	1	1	1
400	1	3	0,333333333
800	1	13	0,076923077
1600	1	51	0,019607843
3200	1	228	0,004385965
6400	1	935	0,001069519
12800	2	4126	0,000484731
25600	3	18480	0,000162338
51200	6	FDT	FDT

Razone si los diferentes tiempos y su cociente concuerda con lo esperado.

En efecto, el comportamiento es el esperado. Para poder reflejar el crecimiento de la tabla, se ha sustituido los tiempos de 0 por 1. Como se observa, el tiempo del bucle2 crece mucho más rápido que el del

bucle 1, por lo que la tabla T1/T2 refleja un número que se aproxima cada vez más a 0.

EJERCICIO 4 –

n	TBucle3	TBucle2	T3/T2
100	1	1	1
200	4	1	4
400	15	3	5
800	64	13	4,923076923
1600	263	51	5,156862745
3200	1100	228	4,824561404
6400	4658	935	4,981818182
12800	19796	4126	4,797867184
25600	FDT	18480	FDT
51200	FDT	FDT	FDT

Razone si los diferentes tiempos y su cociente concuerda con lo esperado.

Ambos bucles tienen la misma complejidad $O(n^2 * \log n)$; pero los tiempos de Bucle2 son menores que los del Bucle3 debido a lo discutido previamente. La diferencia de tiempos, es por ende, una constante multiplicativa, que se aproxima alrededor de 5.

EJERCICIO 5 –

n	TBucle4 PY	TBucle4 sin OP	PY/JAVA
100	3	1	3
200	29	6	4,833333333
400	222	43	5,162790698
800	1597	324	4,929012346
1600	11753	2567	4,578496299
3200	FDT	20506	FDT
6400	FDT	FDT	FDT
12800	FDT	FDT	FDT
25600	FDT	FDT	FDT
51200	FDT	FDT	FDT

Razone si los diferentes tiempos y sus cocientes concuerda con lo esperado.

Los tiempos son los esperados, y sus cocientes también. Ambos bucles son implementaciones del mismo algoritmo, la única diferencia siendo el lenguaje en el que se compila. En concreto, Python va más lento específicamente porque es un lenguaje interpretado, tarda más porque va ejecutando y leyendo el código línea a línea. La diferencia, es de nuevo, una constante multiplicativa entre ambos que se aproxima a 5.