

# **PREGUNTAS SECCIÓN 1.2**

## **Sesión práctica 3**

Jonathan Arias Busto

UO283586

71780982-Y

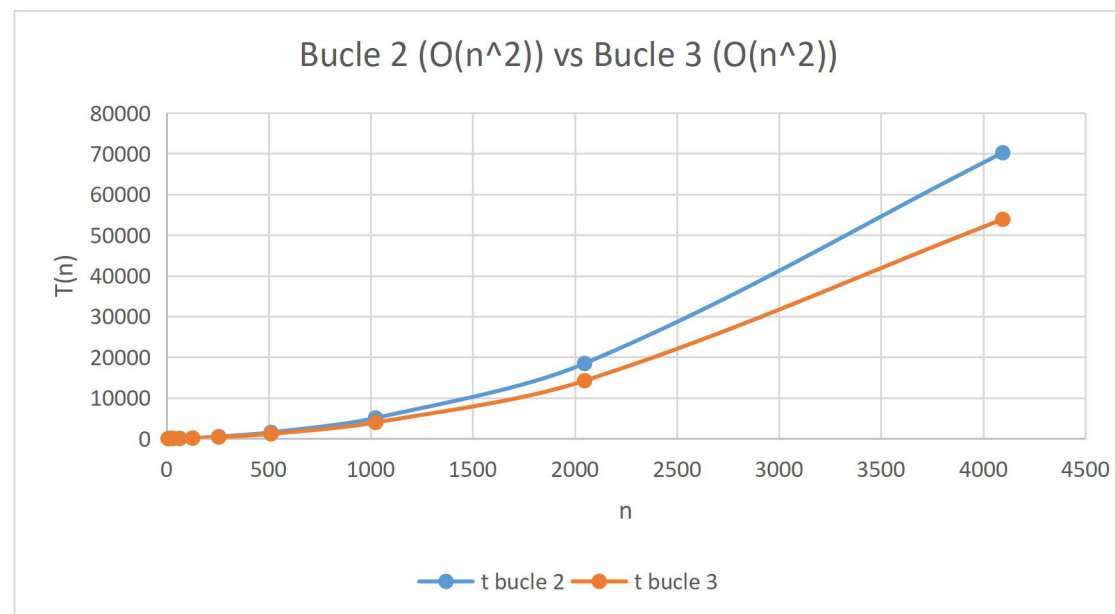
Escuela de Ingeniería Informática - EII

## TABLAS TIEMPOS DIFERENTES ALGORITMOS

En esta práctica se miden tiempos de los diferentes algoritmos implementados, con complejidades como  $O(n^2)$ ,  $O(n^3)$ ,  $O(n^4)$  o  $O(n^3 \log n)$ .

En esta tabla se usa un  $N^\circ$  veces = 100000, por lo que estaríamos hablando de  $100000 = 1ms$

TABLA 1	N	t bucle 2	t bucle 3	t bucle2/bucle3
	8	6	5	1,2
	16	6	5	1,2
	32	19	11	1,727272727
	64	48	30	1,6
	128	113	118	0,957627119
	256	491	369	1,330623306
	512	1527	1164	1,31185567
	1024	5071	3954	1,282498735
	2048	18459	14225	1,297644991
	4096	70218	53861	1,303689126
				1,305980791



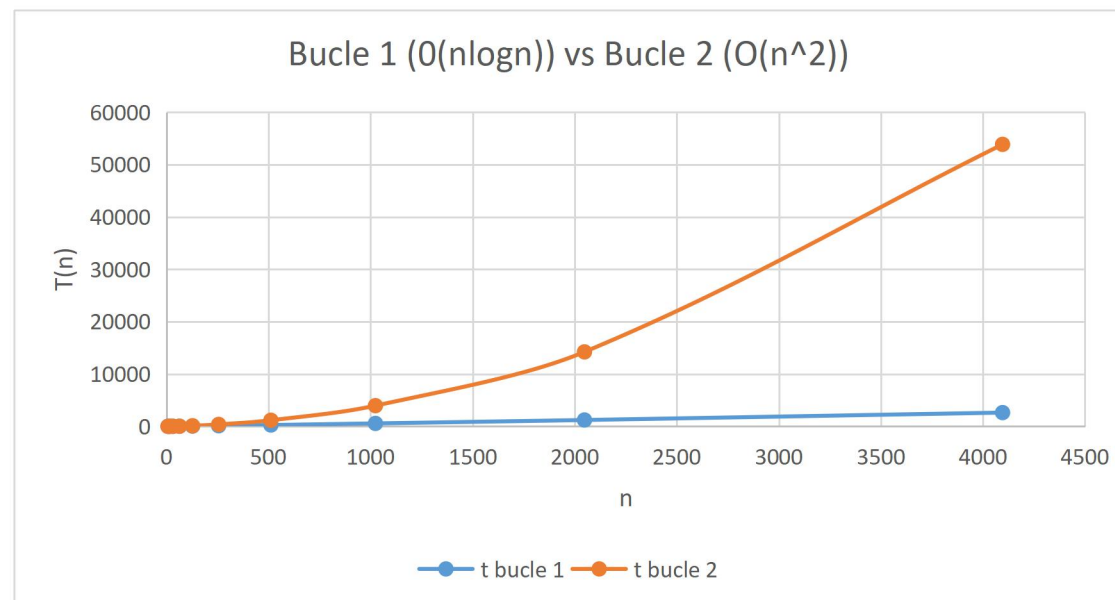
En este caso estamos comparando el tiempo de ejecución del algoritmo de la clase Bucle2 con la clase Bucle3.

En el caso del Bucle2 estamos hablando de una complejidad  $O(n^2)$  clásico. En el caso del Bucle3 también tenemos una complejidad  $O(n^2)$ , pero en este caso la  $j$  del segundo for depende de la variable  $i$  del primer for, por lo que en teoría debería de ejecutarse más rápido. Esto se cumple como se puede apreciar en la gráfica.

El resultado para este caso como podemos comprobar es el esperado

En esta tabla se usa un  $N^{\circ}$  veces = 100000, por lo que estaríamos hablando de 100000 = 1ms

TABLA 2	N	t bucle 1	t bucle 2	t bucle2/bucle3
	8	4	5	0,8
	16	4	5	0,8
	32	11	11	1
	64	23	30	0,766666667
	128	53	118	0,449152542
	256	118	369	0,319783198
	512	260	1164	0,223367698
	1024	564	3954	0,142640364
	2048	1213	14225	0,085272408
	4096	2629	53861	0,048810828
				0,314095964



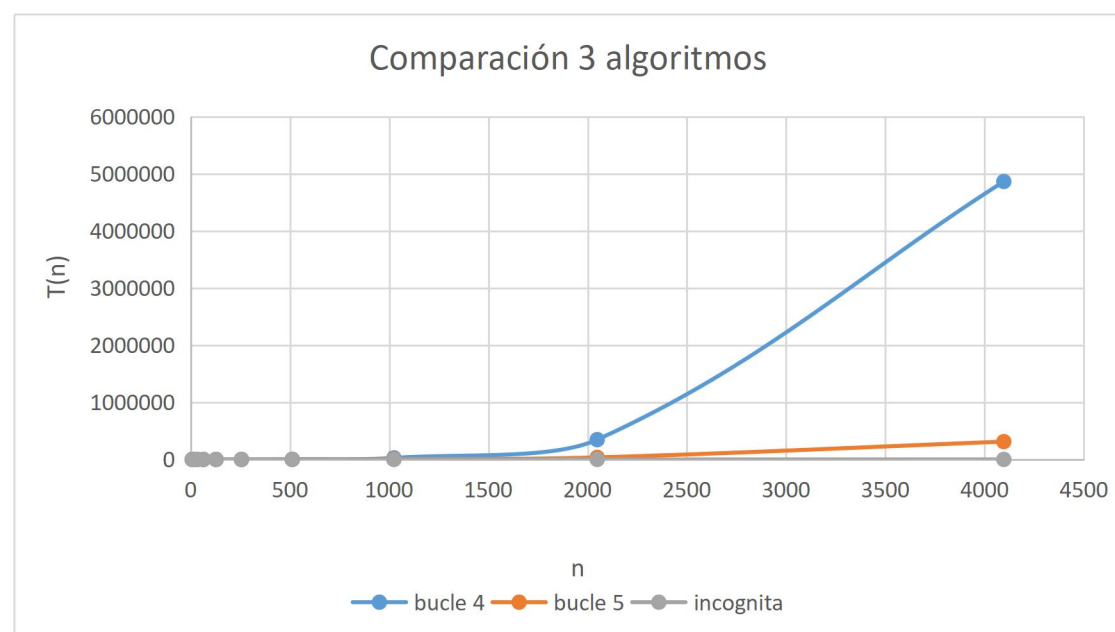
En este caso estamos comparando la complejidad de la clase Bucle1 con la clase Bucle2.

Como vimos en la anterior tabla Bucle 2 tiene una complejidad  $O(n^2)$  debido a sus dos for anidados. En el caso de Bucle1 se trata de una complejidad  $O(n \log n)$ .

Se puede apreciar como la complejidad  $O(n \log n)$  de Bucle 1 es mucho mas rápida que la complejidad del Bucle2. En la gráfica se puede observar como la linea naranja (Bucle2) aumenta de forma exponencial, mientras la linea azul (Bucle1) sigue un crecimiento linea.

El resultado para esta tabla también es el esperado si aplicamos la teoría vista en clase.

TABLA 3				
	n	t bucle 4	t bucle 5	t incognita
	8	0	0	0,001
	16	1	0	0,003
	32	2	2	0,002
	64	9	3	0,008
	128	14	6	0,045
	256	175	56	0,306
	512	2345	457	2,11
	1024	27137	4157	14,157
	2048	346763	35961	100,176
	4096	4864295	312161	752,492
		N veces = 1	N veces = 1	N veces = 1000 (resultados pasados a ms para poder comparar)
		$O(n^4)$	$O(n^3 \log n)$	$O(n^3)$



En este caso, la tabla es algo mas compleja ya que estamos comparando 3 algoritmos diferentes.

Los tres algoritmos son los pertenecientes a las clases Bucle4, Bucle5 e Incognita. Estas clases tienen una complejidad de  $O(n^4)$ ,  $O(n^3 \log n)$  y  $O(n^3)$  respectivamente.

Como se puede apreciar en la gráfica y siguiendo el razonamiento anterior, la complejidad más lenta es  $O(n^4)$  seguida por la complejidad  $O(n^3 \log n)$  y finalmente la más rápida en ejecutar será la complejidad  $O(n^3)$ . Estos resultados son los esperados puesto que la duda estaría en que algoritmo, de entre  $O(n^3)$  y  $O(n^3 \log n)$ , se ejecutaría más rápido. Aplicando algo de matemática y de teoría la complejidad  $O(n^3)$  debería de ser mas rápida debido al factor de la multiplicación por el logn que hará que la función crezca mas rápido que  $O(n^3)$ .

Las gráficas individuales correspondientes a los diferentes algoritmos son las siguientes:

