Algorithmics	Student information	Date	Number of session
	UO: 271580	16/02/20	1.2
	Surname: Lopez Amado		
	Name: Pablo		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Loop2 vs Loop3

n	loop2 (s)	loop3 (s)	loop2/loop3 (s)
64	0,042		
128	0,17	0,089	1,91011236
256	0,69	0,364	1,895604396
512	2,71	1,364	1,986803519
1024	10,945	5,709	1,917148362
2048	43,767	23,132	1,892054297
4096		89,044	

According to the graph we can interpret that Loop2/Loop3 tends to a constant, 1.9. That is because regarding both algorithms have n^2 complexity, loop3 is faster since it does not iterate always to n at its second loop.

Theoretically:

N1 = 128 t1 = 0.17

N2 = 2048 t2 = ?

N2 ^c / N1 ^c * t1 => N2 ^2 / N1 ^2 * t1 => 2048^2/ 128^2 * 0.17 = **43,52 = t2 (43.767)**

N1 = 128 t1 = 0.17

N2 = ? t2 = 43,767

N2^c = t2/t1 * N1^c => N2 = Sqrt c(t2/t1) * N1 => N2 = Sqrt 2(43.767/0.17)*128 = **2053.8 = N2 (2048)**

Algorithmics	Student information	Date	Number of session
	UO: 271580	16/02/20	1.2
	Surname: Lopez Amado		
	Name: Pablo		

Activity 2. Loop1 vs Loop2

N	loop1 (s)	loop2 (s)	loop1/loop2 (s)
32	0,046	0,109	0,422018349
64	0,079	0,438	0,180365297
128	0,156	1,689	0,092362345
256	0,406	6,987	0,058107915
512	0,906	27,41	0,03305363
1024	2,046	108,621	0,018836137

According to the values seen, we can assure that Loop1/Loop2 tends to 0.

Theoretically:

N1 = 64 t1 = 0.079

N2 = 1024 t2 = ?

f(n2)/f(n1) * t1 => n2logn2/n1logn1 * t1 => 1024*log(1024) / 64*log(64) *0.079 = **2.10 = t2 (2.046)**

N1 = 64 t1 = 0.438

N2 = ? t2 = 27.41

n2^c = t2/t1 * n1^c => n2 = sqrt2(t2/t1) * n1 => n2 = srqt2(27.41/0.438) * 64 => **n2 = 506.28 (512)**

Algorithmics	Student information	Date	Number of session
	UO: 271580	16/02/20	1.2
	Surname: Lopez Amado		
	Name: Pablo		

Activity 3. Loop4 vs Loop5

n	loop4 (s)	loop5 (s)	loop4 (s)/ loop5 (s)
16	22	23	0,956521739
32	300	169	1,775147929
64	5381	1507	3,570670206
128	70970	14753	4,810547007
256	1033527	124628	8,292895657

Despite the first values are not enough reliable (I didn't get higher values), it can be seen that Loop4/Loop5 tends to infinity because Loop4 has an n^4 and Loop5 has an n^3logn complexity.

Theoretically:

N1 = 32 t1 = 169

N2 = 256 t2 = ?

T2 = n2/n1 * t1 => n2^c * log(n2)/ n1^c * log(n1) * t1 =>

= > 256^3 * log(256) / 32^3 * log(32) *169 => **138444 = t2 (124628)**

N1 = 32 t1 = 300

N2 = ? t2 = 1033527

n2^c = t2/t1 * n1^c => n2 = sqrt4(t2/t1) * n1 => n2 = srqt4(1033527/300) * 32 => **n2 = 245.16(256)**

Algorithmics	Student information	Date	Number of session
	UO: 271580	16/02/20	1.2
	Surname: Lopez Amado		
	Name: Pablo		

Activity 4. Unknown

c	n	unknown
69450000	32	0,046
52705000	64	0,274
4104650000	128	1,761
3,2395E+10	256	12,062
2,574E+11	512	89,032

The complexity of the algorithm is n^3 because there are 3 nested loops.

Theoretically:

N1 = 32 t1 = 0,046

N2 = 512 t2 = ?

N2 ^c / N1 ^c * t1 => N2 ^3 / N1 ^3 * t1 => 512^3/ 32^3 * 0.046 = **188,416 = t2 (89,032)**

N1 = 32 t1 = 0,046

N2 = 256 t2 = ?

N2 ^c / N1 ^c * t1 => N2 ^3 / N1 ^3 * t1 => 256^3/ 32^3 * 0.046 = **23,552 = t2 (12,062)**

Both results are higher than expected because whereas the complexity is n^3, each nested loop has only to “arrive” to the number at the previous loop, then the deeper (I,j or k higher) the loop is executed the bigger the difference to the theoretical value.