


Algorithmics	Student information	Date	Number of session
	UO: 300047	03/02/2025	1
	Surname: Rodríguez Torres	 Escuela de Ingeniería Informática Universidad de Oviedo	
	Name: Luisa Natalia		



## Activity 1. Factor 1: problem size

Iterations	Time (milliseconds)
10000	1532
20000	6170
40000	24940
80000	100428
16000	OoT
320000	OoT
640000	OoT

We can see that the Python A1 problem has  $O(n^2)$  complexity. This aligns with the results obtained. The time grows exponentially as the problem size.

## Activity 2. Factor 2: computer performance

Computer 1) CPU: 12<sup>th</sup> Gen Intel® Core™ i5-12400

RAM: 16,0 GB

Computer 2) 12th Gen Intel(R) Core(TM) i5-1235U 2.50 GHz

RAM: 8,0 GB

Iterations	Time C1 (milliseconds)	Time C2 (milliseconds)
10000	1532	2182
20000	6170	7318
40000	24940	2968
80000	100428	OoT

Algorithmics	Student information	Date	Number of session
	UO: 300047	03/02/2025	1
	Surname: Rodríguez Torres		
	Name: Luisa Natalia		

16000	OoT	OoT
320000	OoT	OoT
640000	OoT	OoT

We can see how there is a clear difference in time corresponding to the computers power. For example the second computer only has half the RAM of the first computer, and as such, it is slower.

### Activity 3. Factor 3: implementation environment

Iterations	Time Java (milliseconds)	Time Python (milliseconds)
10000	13	1532
20000	34	6170
40000	133	24940
80000	458	100428
160000	2111	OoT
320000	6589	OoT
640000	24663	OoT

Java is much faster than Python, this is due to structural differences. For example, Java is a static program, which means the variables are known at compile time, while Python is a dynamic program, which means they are known at runtime.

### Activity 4. Factor 4: algorithm that is used

Iterations	Time A1 (ms)	Time A2 (ms)	Time A3 (ms)
------------	--------------	--------------	--------------

Algorithmics	Student information	Date	Number of session
	UO: 300047	03/02/2025	1
	Surname: Rodríguez Torres		
	Name: Luisa Natalia		

10000	1532	186	94
20000	6170	677	352
40000	24940	2532	1295
80000	100428	9511	4872
16000	OoT	37467	18341
320000	OoT	OoT	OoT
640000	OoT	OoT	OoT

All of the algorithms present a different complexity. A1 is  $O(n^2)$ , A2 is  $O(n \cdot \sqrt{n})$ , and A3 is  $O(n \cdot \log(n))$ . As we can see, each of these complexities is better than the last, so even if we have the same problem, the algorithm used can make times much faster.