

Algorithmics	Student information	Date	Number of session
	UO: 283928	03/29	Session 5
	Surname: Suárez Losada		
	Name: Gonzalo		

Activity 1. Execution times of the algorithm.

Size(chars)	Time(ms)
200	3
400	3
800	8
1600	13
3200	23
6400	97
12800	404

The complexity of this algorithm is $O(n \cdot m)$, where n and m are the length of the first and the second strings, respectively. In our very case, as both strings have the same length, the complexity can be simplified as $O(n^2)$.

Now, let us compute if the execution times resemble to the estimated values:

$$T_2 = T_1 \cdot f(n_2) / f(n_1), \text{ where } k^c = n_2 / n_1$$

Then:

$$T_2 = 4 \cdot T_1$$

That is, as it is an algorithm which complexity is quadratic, for workloads large enough, each execution time will be, approximately, four times the previous case.

Can be easily seen on the table that this is hold from size $n = 3200$ with almost no error.

Prove on the correct execution of the algorithm:

```
<terminated> LeveshteinDistanceTest [Java Application] C
Size of Strings: 9 Time=0
  A B R A C A D A B R A
0 1 2 3 4 5 6 7 8 9 10 11
B 1 1 1 2 3 4 5 6 7 8 9 10
A 2 1 2 2 2 3 4 5 6 7 8 9
R 3 2 2 2 3 3 4 5 6 7 7 8
C 4 3 3 3 3 3 4 5 6 7 8 8
A 5 4 4 4 3 4 3 4 5 6 7 8
Z 6 5 5 5 4 4 4 4 5 6 7 8
A 7 6 6 6 5 5 4 5 4 5 6 7
S 8 7 7 7 6 6 5 5 5 5 6 7
```

As seen in lab sessions with this very example, the result is the same(7).