Algorithmics	Student information	Date	Number of session
	UO: UO269412		1-1
	Surname: Carrillo		
	Name: Javier		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Measuring execution times

1. How many more years can we continue using this way of counting?

If we calculate the substraction of the maximum value the long type provides minus the current time in milliseconds of the system, to later pass such data to years using conversion factors, we obtain a value of 292471158 years until the currentTimeMilis() method stops working.

2. What does it mean that the time measured is 0?

Since it takes so little time to execute the method (it takes a time like 0,00..00XX ms), when we display it in ms, it is only shown as 0, because of such a low value.

3. From what size of problem (n) do we start to get reliable times?

100000000. A smaller value would suppose getting values of time like 0 ms or a low time like 4ms, which we should not consider. With this value of n, the first value we obtain is 41 ms, which can be considered.

Activity 2. Grow of the problem size

1. What happens with time if the size of the problem is multiplied by 5?

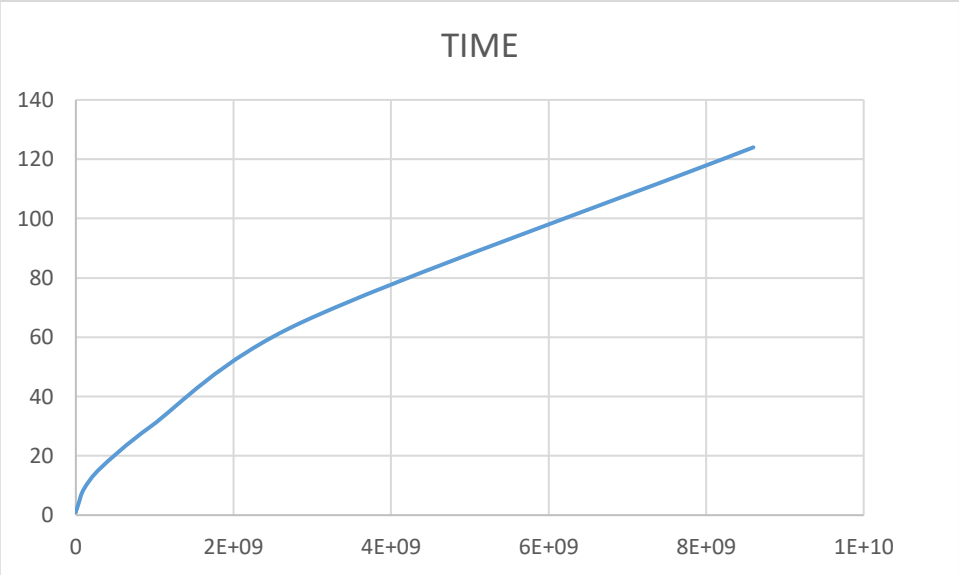
Since the size was increased, it also takes more time to compute the method with a larger sample.

2. Are the times obtained those that were expected from linear complexity O(n)?

Yes, the pattern of values tends to follow a straight line, even if at the beginning it looks a little bit curved (probably due to start up or background programs)

Algorithmics	Student information	Date	Number of session
	UO: UO269412		1-1
	Surname: Carrillo		
	Name: Javier		

3. Use a spreadsheet to draw a graph with Excel. On the X axis we can put the time and on the Y axis the size of the problem



Activity 3. Taking small execution times

n	fillIn(10^-5 seconds)	sum(10^-6 seconds)	maximum(10^-6 seconds)
10	28	16	31
30	47	16	46
90	110	62	47
270	297	188	157
810	906	593	484
2430	2717	1813	1422
7290	8106	5564	4251
21870	24025	16485	14063
65610	72130	49012	42189
196830	217019	150381	134784

- What are the main components of the computer in which you did the work (process, memory)?

Algorithmics	Student information	Date	Number of session
	UO: UO269412		1-1
	Surname: Carrillo		
	Name: Javier		

The CPU has 4 cores and 4 logic processors. The memory consisted of a 4GB RAM.

- Do the values obtained meet the expectations? For that, you should calculate and indicate the theoretical values (a couple of examples per column) of the time complexity. Explain briefly the results.

We will calculate the theorical t2, having picked a n1, n2 and t1 from the table. In this case, we will pick n1 = 270, n2 = 810, and t1 of the different n1 of the methods. First, we will begin by calculating the t2 of the fillin method, thus t1 = 297. To calculate t2 we use the formula t2 = n2/n1 * t1 since all three methods are linear. With the provided data we obtain that t2 = 891. There's little difference in comparison with the real value obtained (906). We repeat the same process with the sum method and maximum, using the same n1 and n2 as before. The theorical t2 of sum is 564 which is close to the 593 obtained. The theorical t2 of maximum is 471, which again is close to the real time which was 484.

Trying with other pair of n1 and n2, we will use n1 = 2430 and n2 = 7290. In the fillin method we obtain a t2 of 8151 which is close to the real 8106. In the sum method the we find that the theorical t2 is 5439, again, really close to the real value of 5564. To end up, we have that the t2 of maximum is 4266, close to the real 4266.

Thus, we can conclude that the results indeed meet the expectations.

Activity 4. Operations on matrices

n	sumDiagonal1(10^-5 seconds)	sumDiagonal2(10^-6 seconds)
10	72	14
30	328	25
90	3812	62
270	26458	287
810	214919	1521
2430		20597

Algorithmics	Student information	Date	Number of session
	UO: UO269412		1-1
	Surname: Carrillo		
	Name: Javier		

7290		86672

- What are the main components of the computer in which you did the work (process, memory)?

The RAM memory consisted of 8 GB, while the CPU has 4 cores and 4 logic processors.

- Do the values obtained meet the expectations? For that, you should calculate and indicate the theoretical values (a couple of examples per column) of the time complexity. Explain briefly the results.

We will repeat the same procedure as before, picking n1 and n2, and the t1 obtained in the experiment and calculating the theorical t2 for such data.

In this first round we will take n1 = 30 and n2 = 90, and the t1 for sumDiagonal1= 328 and so the t2 for sumDiagoanl2 = 25. In this case we need to take in account that sumDiagonal1 is quadratic, while sumDiagonal2 is linear, thus the procedure for obtaining t2 will be different in each other (being t2 = n2^2/n1^2 * t1 in the former and t2 = n2/n1 * t1 in the latter). After operating we obtain a t2 = 2952 in the first method and a t2 = 72 in the second method, values close to the real values obtained (3812, 62)

In the second round we will follow the same procedure but taking n1 = 270 and n2 = 810. We obtain a theorical t2 of sumDiagonal1 of 238122 (close to the real 214919); and a theorical t2 of sumDiagonal2 of 861 which differs a lot from the wanted 1521.

We can conclude that sumDiagonal1 meets the expectations but something with sumDiagonal2 went wrong (maybe cpu was busy?), even though the first round of theorical values were close to the real ones. This experiment was repeated several times, but the times keep being different from the theorical.

Activity 5. Benchmarking

1. Why you get differences in execution time between the two programs?

This is due to the compiling of Java, which compared to the way Python interprets the code, it is much faster and thus it provides better results when executing the methods.

2. Regardless of the specific times, is there any analogy in the behavior of the two implementations?

Yes, since both programs implement the same methods (in their proper syntax), we can see that although the times differ from each other, they tend to follow a pattern according to the complexity of the methods (linear, quadratic)