Algorithmics	Student information	Date	Number of session
	UO:269546	10-02-21	1
	Surname: Fernández Arias		
	Name:Sara		

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?

Since long type elements have a maximum value of ,(2^63)-1=9.223.372.036.854.775.807. Provided that each year has 31536000000ms , this method could be used for (rounding it up) 292.471.208,67 years. Provided that it's been 50 years since it's creation, we could tell that we can use it for (more or less) 292.471.158 years more

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

It means that the process was so quick there's almost no difference between the initial and the end measurements , so It gets rounded up to 0.

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

After many attempts , I got the following results:

70.000.000-> 51 ms

69.000.000 -> 54 ms

(I'm aware that measurements may vary slightly from one measure to another.)

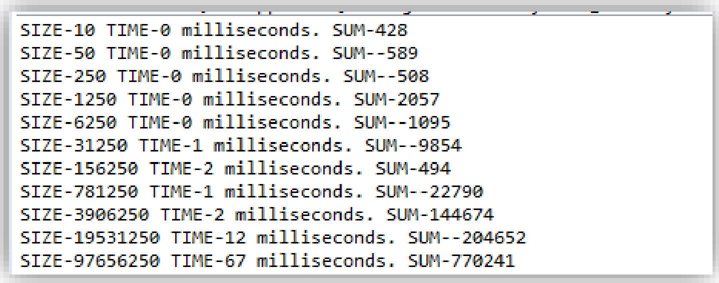
Therefore, I think that reliable measurements are obtained from sizes of 70 million in advance.

Activity 2. Growth of the problem size

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

   The time taken to execute the program will grow proportionally to the size of the problem, therefore , as the size increases five times per iteration; the time will be increased as well proportionally (Since the problem has a linear complexity).

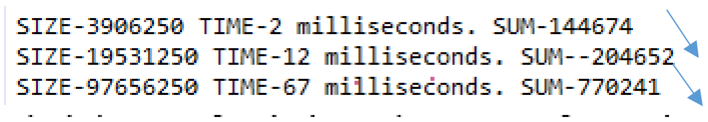
2. Are the times obtained those that were expected from linear complexity O(n)?



```
SIZE-10 TIME-0 milliseconds. SUM-428
SIZE-50 TIME-0 milliseconds. SUM--589
SIZE-250 TIME-0 milliseconds. SUM--508
SIZE-1250 TIME-0 milliseconds. SUM-2057
SIZE-6250 TIME-0 milliseconds. SUM--1095
SIZE-31250 TIME-1 milliseconds. SUM--9854
SIZE-156250 TIME-2 milliseconds. SUM-494
SIZE-781250 TIME-1 milliseconds. SUM--22790
SIZE-3906250 TIME-2 milliseconds. SUM-144674
SIZE-19531250 TIME-12 milliseconds. SUM--204652
SIZE-97656250 TIME-67 milliseconds. SUM-770241
```

	Student information	Date	Number of session
Algorithmics	UO:269546	10-02-21	1
	Surname: Fernández Arias		
	Name:Sara		

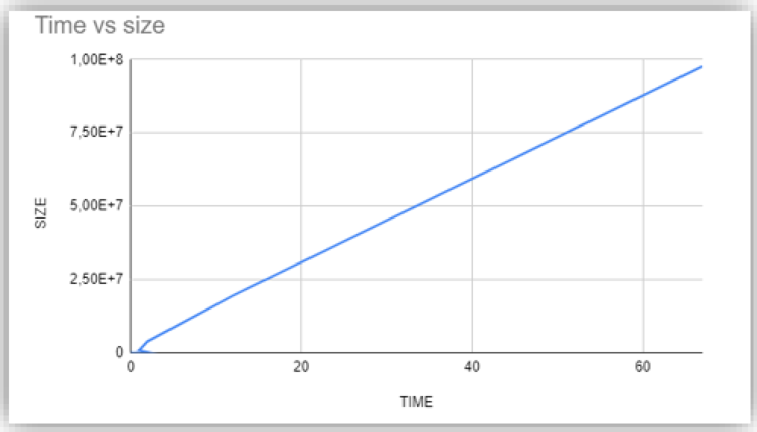
I think they are. If we look into the most significant values (taking into account they aren’t actually significant measuring-wise, but they are the greatest obtained), we can see the proportion:



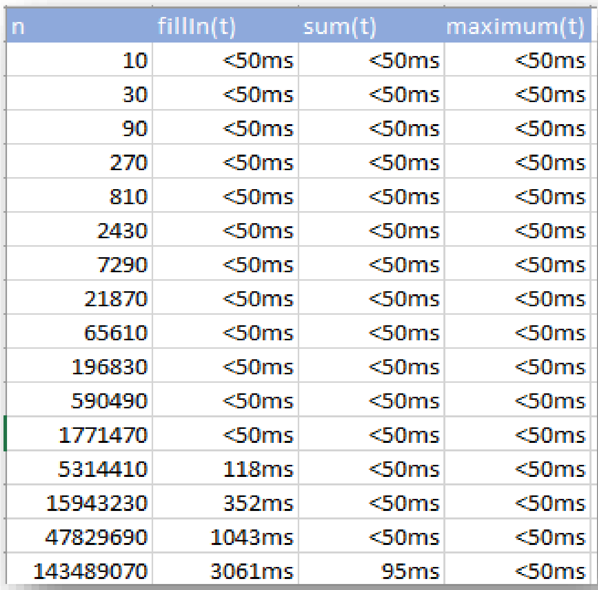
```
SIZE-3906250 TIME-2 milliseconds. SUM-144674
SIZE-19531250 TIME-12 milliseconds. SUM--204652
SIZE-97656250 TIME-67 milliseconds. SUM-770241
```

2*5=10 ms ~ 12 ms and then 12*5=60ms ~67 ms (the output it’s rounded but decimals might change the result slightly)

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

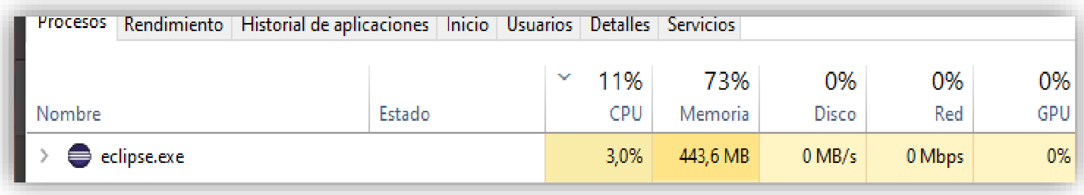


n	fillIn(t)	sum(t)	maximum(t)
10	<50ms	<50ms	<50ms
30	<50ms	<50ms	<50ms
90	<50ms	<50ms	<50ms
270	<50ms	<50ms	<50ms
810	<50ms	<50ms	<50ms
2430	<50ms	<50ms	<50ms
7290	<50ms	<50ms	<50ms
21870	<50ms	<50ms	<50ms
65610	<50ms	<50ms	<50ms
196830	<50ms	<50ms	<50ms
590490	<50ms	<50ms	<50ms
1771470	<50ms	<50ms	<50ms
5314410	118ms	<50ms	<50ms
15943230	352ms	<50ms	<50ms
47829690	1043ms	<50ms	<50ms
143489070	3061ms	95ms	<50ms

Algorithmics	Student information	Date	Number of session
	UO:269546	10-02-21	1
	Surname: Fernández Arias		
	Name:Sara		

**The values where obtained using nTimes=1000*

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



Procesos	Rendimiento	Historial de aplicaciones	Inicio	Usuarios	Detalles	Servicios

Nombre	Estado	11% CPU	73% Memoria	0% Disco	0% Red	0% GPU
eclipse.exe		3,0%	443,6 MB	0 MB/s	0 Mbps	0%

The measurements were made using an Intel® Core ™ i7-8550U CPU and a 8 GB RAM.
The main components where the work is done are the memory and the CPU, specially the memory.

2-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.

Using the following logic:

$$\boldsymbol{T_2 = {f(n_2)}/{f(n_1)}} \qquad \boldsymbol{f(n) = O(n^C) \Rightarrow T_2 = k^C \cdot T_1} \qquad \boldsymbol{k = n_2/_{n_1}}$$

<u>Fill in values</u>

For a size n1=5314410 , it took t1=118ms being k=3.

For a size of 15943230(n2) it will take t2= k^C* t1= $3^1 \cdot 118ms$=354ms.

Empirically , 352ms were obtained. That means it's behaving as expected.

For a size n1=15943230 , it took t1=354ms.

For a size of 47829690 (n2) it will take t2= k^C* t1= $3^1 \cdot 354\text{ms.}$= 1.062

Empirically , 1042ms were obtained. That means it's behaving as expected.

<u>Sum values</u>

For a size n1= 143489070, it took t1=95ms, being k=3.

For a size of 430.467.210 (n2) it will take t2= k^C* t1= $3^1 \cdot 95ms$=354ms.

Algorithmics	Student information	Date	Number of session
	UO:269546	10-02-21	1
	Surname: Fernández Arias		
	Name:Sara		

Empirically , 352ms were obtained. That means it's behaving as expected.

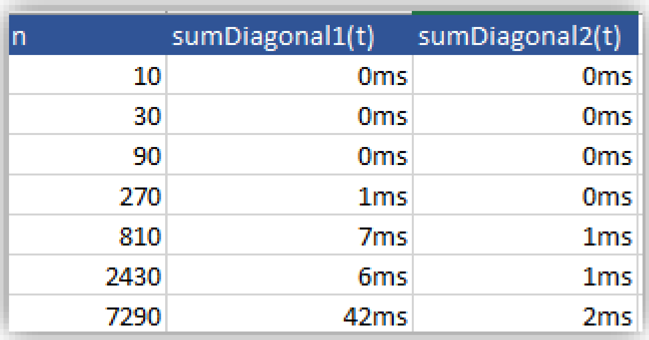
For a size n1=15943230 , it took t1=354ms. being k=3.

For a size of 47829690 (n2) it will take t2= k^C* t1= $3^1 \cdot 354\text{ms.}$= 285ms
Empirically , the only value obtained that was significant before the crash of the execution , was 95ms, so I have no empirical values to compare it with.

Maximum values

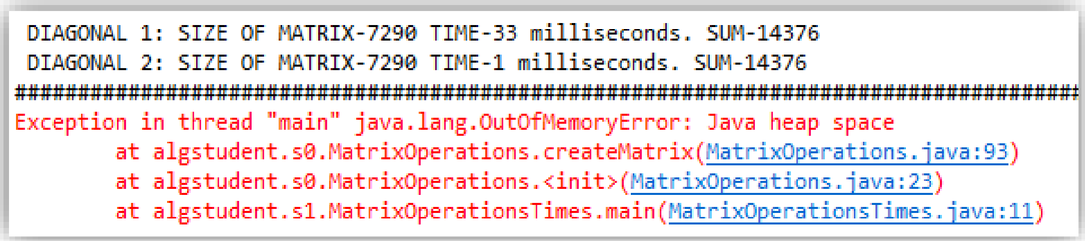
No significant values where obtained before the crash of the execution, meaning the performance of this program is really good.

Activity 4. Operations on matrices



n	sumDiagonal1(t)	sumDiagonal2(t)
10	0ms	0ms
30	0ms	0ms
90	0ms	0ms
270	1ms	0ms
810	7ms	1ms
2430	6ms	1ms
7290	42ms	2ms

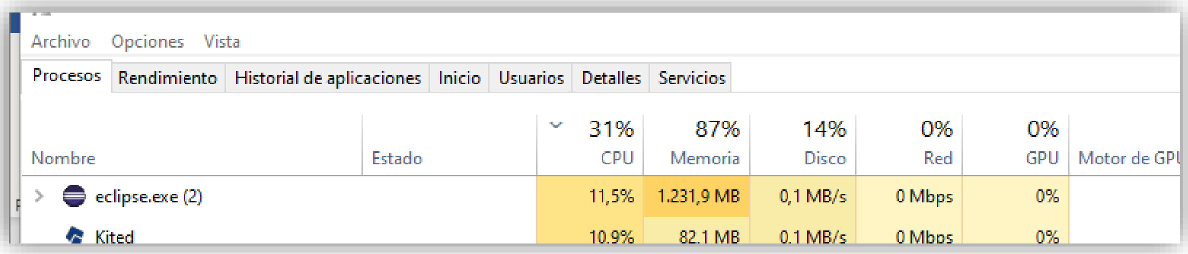
It crashes when n=7290



```
 DIAGONAL 1: SIZE OF MATRIX-7290 TIME-33 milliseconds. SUM-14376
 DIAGONAL 2: SIZE OF MATRIX-7290 TIME-1 milliseconds. SUM-14376
###################################################################################
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
	at algstudent.s0.MatrixOperations.createMatrix(MatrixOperations.java:93)
	at algstudent.s0.MatrixOperations.<init>(MatrixOperations.java:23)
	at algstudent.s1.MatrixOperationsTimes.main(MatrixOperationsTimes.java:11)
```

Algorithmics	Student information	Date	Number of session
	UO:269546	10-02-21	1
	Surname: Fernández Arias		
	Name:Sara		

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



Nombre	Estado	31% CPU	87% Memoria	14% Disco	0% Red	0% GPU	Motor de GPU
eclipse.exe (2)		11,5%	1.231,9 MB	0,1 MB/s	0 Mbps	0%	
Kited		10,9%	82,1 MB	0,1 MB/s	0 Mbps	0%	

The measurements were made using an Intel® Core ™ i7-8550U CPU and a 8 GB RAM. Again , both memory and the CPU are being used. Also the disk, but mainly CPU and memory.

2-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.

No significant values where obtained from the execution of this program.

Activity 5. Benchmarking

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

The only difference I see that must be making such a difference is that the python version is run using an interpreter, that translates the program one statement at a time. Generally speaking, interpreters are faster than compilers.

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

The analogy between both is that the greater the complexity , the longer it takes. In both executions , the linear implementation is faster than the quadratic one.