Algorithmics	Student information	Date	Number of session
	UO: UO282276	15/02/2022	2
	Surname: Cadenas Blanco		
	Name: Andrés		

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?

It can continue from now till 292.471.156,5171606 years.

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

It means that the computer runs the algorithm so fast that it cannot reflect it with such a small number. For that the number of times should be increased until we get a reliable time

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

As I'm using at the moment the laptop in power mode, and it is quite potent I had to use a really big power of ten:

```
SIZE = 1000000000 - TIME = 364 milliseconds SUM = 541054
```

In fact, that n divided by 10 would result only in 30ms.

Activity 2. Grow of the problem size

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

Firstly, with "small" n values as always, the time is 0 or nearly 0 but suddenly it starts to increase exponentially until it crashes.

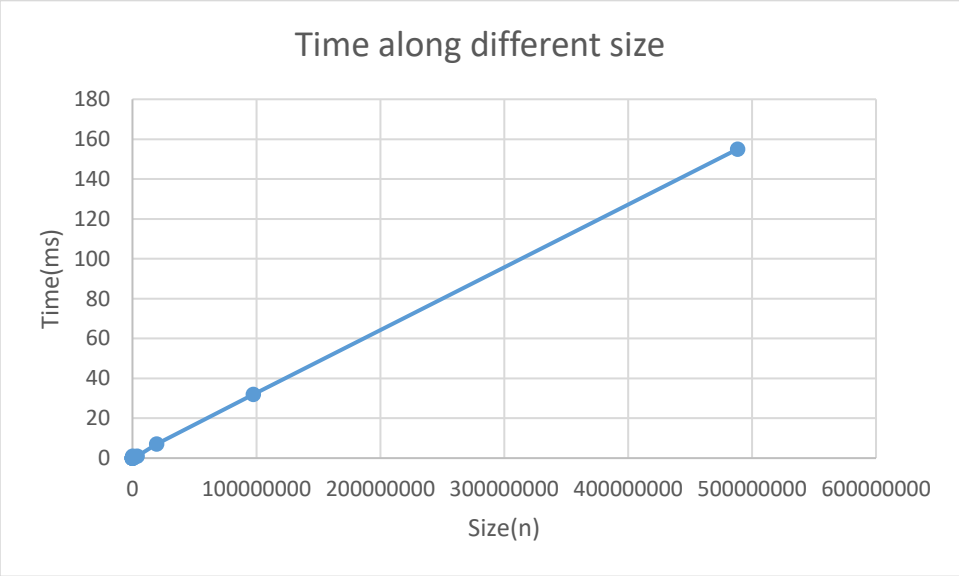
2. Are the times obtained those that were expected from linear complexity O(n)?

Assuming some milliseconds of error we can agree that the results are as expected. For example, as each step is 5 times n the one before when you multiply the value obtained of 32 ms by 5 you get 160 ms and the actual result was 155 ms.

Algorithmics	Student information	Date	Number of session
	UO: UO282276	15/02/2022	2
	Surname: Cadenas Blanco		
	Name: Andrés		

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.

DISCLAIMER: I used the axis inversed because excel wouldn't allow me to do it the other way



It is obvious that the complexity is O(n).

Algorithmics	Student information	Date	Number of session
	UO: UO282276	15/02/2022	2
	Surname: Cadenas Blanco		
	Name: Andrés		

Activity 3. Taking small execution times

You should use the previous concepts for the three following methods: fillIn(), sum() and maximum(). With the values obtained, you should complete the following table:

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

Mostly in cpu and RAM memory, but the hardwork was done by the cpu.

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. Briefly explain the results.

N	fillIn()	sum()	maximum()
10	5	0	1
30	4	0	1
90	9	0	1
270	25	0	1
810	75	2	3
2430	212	7	6
7290	633	18	2
21870	1905	53	72
65610	5686	161	221
196830	16930	479	701

fillIn() method: Meets the theoretical values as taking th time of 75 ms and the constant being 3, 25*3 =75 ms and 75*3 = 225 which is near to 212.

sum() method: It also meets the conditions as the constant is the same and 53*3 = 159

maximum() method: I wont do any calculation but it has the same proportions approximately, therefore it is reliable.

Algorithmics	Student information	Date	Number of session
	UO: UO282276	15/02/2022	2
	Surname: Cadenas Blanco		
	Name: Andrés		

Activity 4. Operations on matrices

N	sumDiagonal1	sumDiagonal2
10	0	0
30	3	0
90	1	0
270	18	0
810	152	1
2430	1294	16
7290	11673	59
21870	104262	272

The repetitions were set to 1000 so there would be representative values

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

The main usage is cpu and memory, in fact it crashed because of the memory used.

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
        at algstudent.s0.MatrixOperations.<init>(MatrixOperations.java:17)
        at algstudent.s11.MatrixOperationsTimes.timeOfSumDiagonal2(MatrixOperationsTimes.java
        at algstudent.s11.MatrixOperationsTimes.main(MatrixOperationsTimes.java:12)
```

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. Briefly explain the results.

Yes they do, as sumDiagonal1 has complexity n^2 the constant in this case is 3^2 that is 9 and taking 152 it matches 1294 approximately when multiplied by the constant whereas sumDiagonal2 is linear and it happens the same. The formula is:

$$t2 = \frac{n2}{n1} * t1$$

Algorithmics	Student information	Date	Number of session
	UO: UO282276	15/02/2022	2
	Surname: Cadenas Blanco		
	Name: Andrés		

Activity 5. Benchmarking

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

Because each programming language has different times at execution, python is known to be way slower than C for example.

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

The code is almost the same however due to the difference of the languages the python variables are dynamically selected the type and an import must be done to take the time. It is probably there where the times are increased.