

Mikel Fernández Esparta

UO275688

Algorithmics

LAB I-3

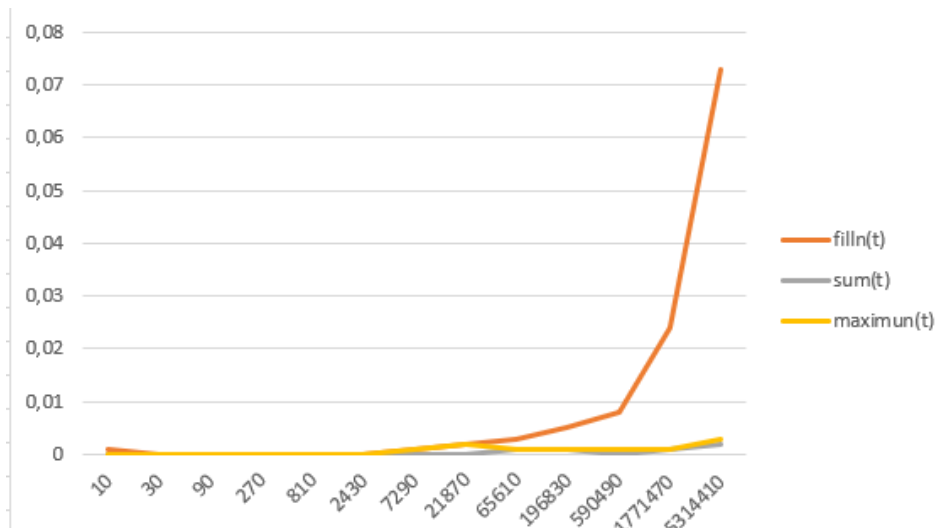
Activity 1: Measuring execution times

1. **How many more years can we continue using this way of counting?**
2.922710230453132×10⁻¹⁰ years, since it is the maximum value for a long type.
2. **What does it mean that the time measured is 0?**
0 appears when you call System.currentTimeMillis() twice and you get the exact same number twice. Otherwise, the result will be the difference in time needed to perform a task.
3. **From what size of problem (n) do we start to get reliable times?**
From 100000, if not the values would be 0.

Activity 2: Grow of the problem size

1. **What happens with time if the size of the problem is multiplied by 5?**
The execution time is slightly higher.
2. **Are the times obtained those that were expected from linear complexity O(n)?**
Yes.
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.**

n	filln(t)	sum(t)	maximun(t)
10	0,001 milliseconds	0,000 milliseconds	0,000 milliseconds
30	0,000 milliseconds	0,000 milliseconds	0,000 milliseconds
90	0,000 milliseconds	0,000 milliseconds	0,000 milliseconds
270	0,000 milliseconds	0,000 milliseconds	0,000 milliseconds
810	0,000 milliseconds	0,000 milliseconds	0,000 milliseconds
2430	0,000 milliseconds	0,000 milliseconds	0,000 milliseconds
7290	0,001 milliseconds	0,000 milliseconds	0,001 milliseconds
21870	0,002 milliseconds	0,000 milliseconds	0,002 milliseconds
65610	0,003 milliseconds	0,001 milliseconds	0,001 milliseconds
196830	0,005 milliseconds	0,001 milliseconds	0,001 milliseconds
590490	0,008 milliseconds	0,000 milliseconds	0,001 milliseconds
1771470	0,024 milliseconds	0,001 milliseconds	0,001 milliseconds
5314410	0,073 milliseconds	0,002 milliseconds	0,003 milliseconds



Activity 3: Taking small execution times

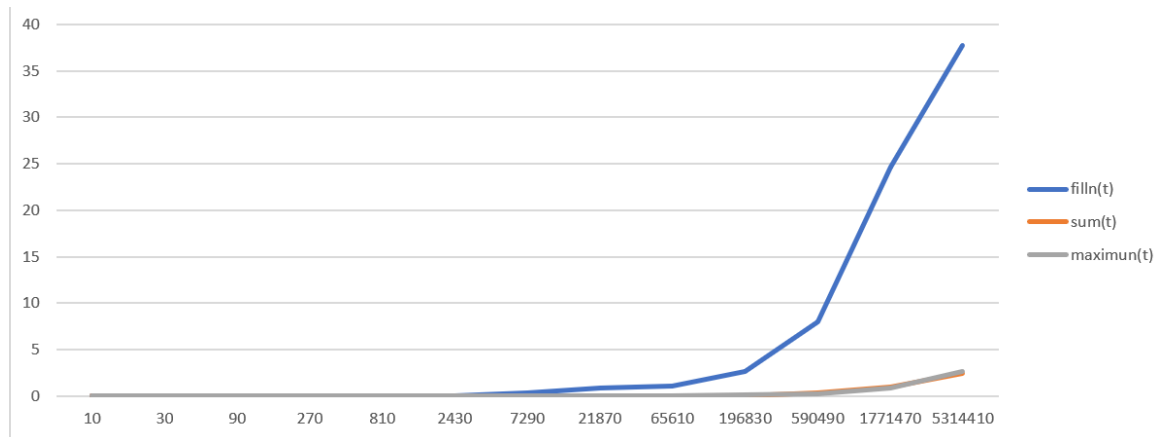
What are the main components of the computer in which you did the work (process, memory)?

Process

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.

Yes, the complexity is $O(n^3)$

n	filln(t)	sum(t)	maximun(t)
10	0,006 milliseconds	0,0 milliseconds	0,001 milliseconds
30	0,001 milliseconds	0,0 milliseconds	0,001 milliseconds
90	0,002 milliseconds	0,0 milliseconds	0,001 milliseconds
270	0,005 milliseconds	0,001 milliseconds	0,001 milliseconds
810	0,011 milliseconds	0,001 milliseconds	0,001 milliseconds
2430	0,036 milliseconds	0,001 milliseconds	0,002 milliseconds
7290	0,374 milliseconds	0,003 milliseconds	0,003 milliseconds
21870	0,862 milliseconds	0,007 milliseconds	0,009 milliseconds
65610	1,078 milliseconds	0,023 milliseconds	0,027 milliseconds
196830	2,668 milliseconds	0,065 milliseconds	0,082 milliseconds
590490	7,999 milliseconds	0,337 milliseconds	0,249 milliseconds
1771470	24,682 milliseconds	0,927 milliseconds	0,812 milliseconds
5314410	37,745 milliseconds	2,482 milliseconds	2,67 milliseconds



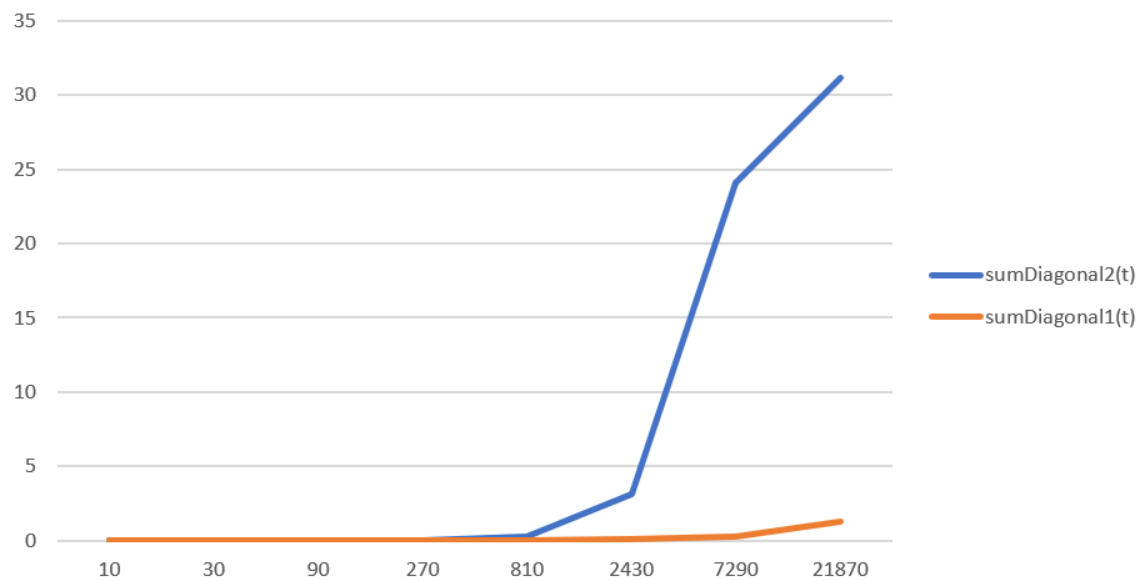
Activity 4: Operations on matrices

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

Process

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.

n	sumDiagonal1(t)	sumDiagonal2(t)
10	0,0 milliseconds	0,001 milliseconds
30	0,0 milliseconds	0,007 milliseconds
90	0,001 milliseconds	0,003 milliseconds
270	0,001 milliseconds	0,025 milliseconds
810	0,012 milliseconds	0,233 milliseconds
2430	0,079 milliseconds	3,086 milliseconds
7290	0,282 milliseconds	24,107 milliseconds
21870	1,26 milliseconds	31,2 milliseconds



Activity 5: Benchmarking

- 1. Why you get differences in execution time between the two programs?**
Although both programs are the same, the values are different because the value of time that is read changes constantly when executing and they are two different languages.
- 2. Regardless of the specific times, is there any analogy in the behavior of the two implementations?**
The code of both programs is basically the same but in a different language.