

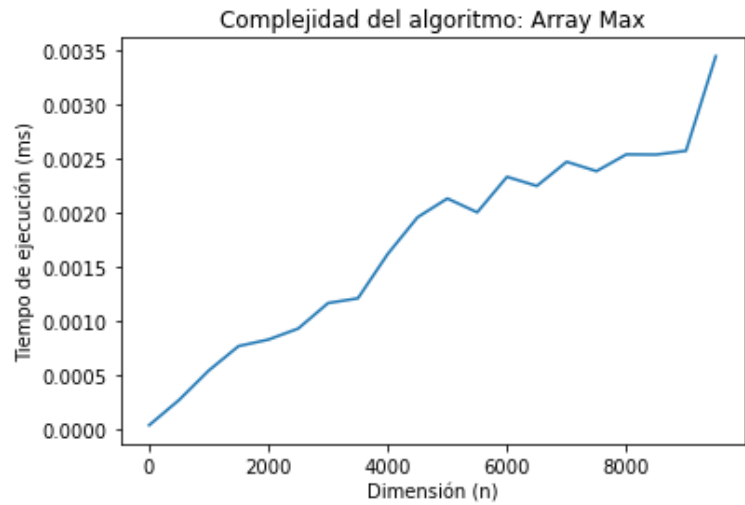
Taller 4: Complejidad

Martin Ospina Uribe y María José Bernal Vélez

Ejercicio 1: Array Max

n	Tiempo ejecución (ms)
100	3.5762786865234375e-05
600	0.00026798248291015625
1100	0.0005407333374023438
1600	0.0007660388946533203
2100	0.0008261203765869141
2600	0.0009279251098632812
3100	0.0011630058288574219
3600	0.0012068748474121094
4100	0.0016160011291503906
4600	0.001954793930053711
5100	0.002128124237060547
5600	0.002000093460083008
6100	0.0023279190063476562
6600	0.0022437572479248047
7100	0.002466917037963867
7600	0.0023801326751708984
8100	0.0025348663330078125
8600	0.002532958984375
9100	0.002567768096923828
9600	0.0034410953521728516

Gráfica de complejidad experimental:



Complejidad teórica:

$$T(n) = c + T(n + 1)$$

$$T(n) = c_1 - cn$$

Notación O:

$$O(n)$$

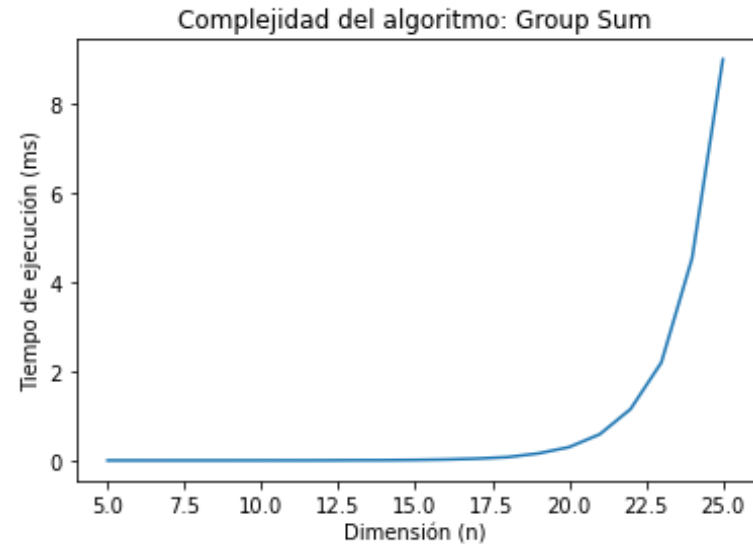
Análisis:

A pesar de que la gráfica anterior no es completamente lineal, muestra una tendencia a serlo, por lo que podemos decir que la complejidad teórica es correspondiente a la complejidad obtenida experimentalmente, pero al tener pocos datos experimentales, no es completamente exacta.

Ejercicio 2: Group Sum

n	Tiempo ejecución (ms)
5	0.00013685226440429688
6	5.626678466796875e-05
7	5.793571472167969e-05
8	0.00010085105895996094
9	0.0001678466796875
10	0.0003597736358642578
11	0.0009319782257080078
12	0.0014908313751220703
13	0.003982067108154297
14	0.0054111480712890625
15	0.010620832443237305
16	0.021265268325805664
17	0.04012012481689453
18	0.07414889335632324
19	0.1571042537689209
20	0.2980082035064697
21	0.5872299671173096
22	1.1518042087554932
23	2.1945130825042725
24	4.533586740493774
25	8.991566896438599

Gráfica de complejidad experimental:



Complejidad teórica:

$$T(n) = c + T(n-1) + T(n-1)$$

$$T(n) = c(2^n - 1) + c_1 2^{n-1}$$

Notación O:

$$O(2^n)$$

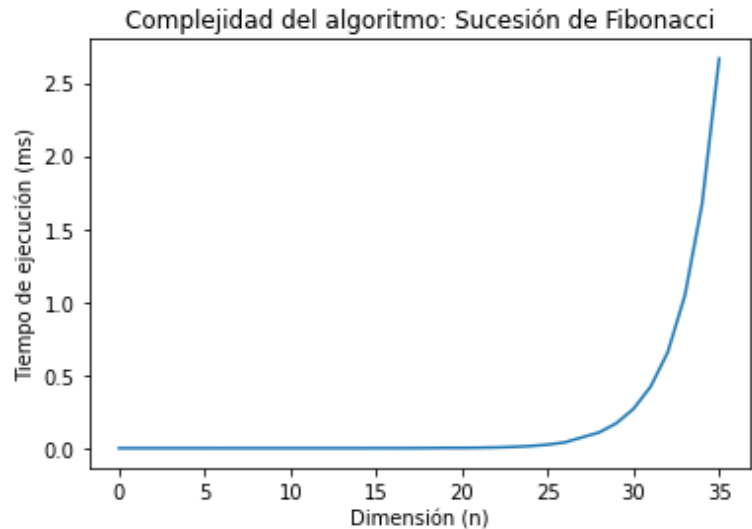
Análisis:

La complejidad teórica corresponde con la complejidad experimental, debido a que en la gráfica podemos ver que el algoritmo para calcular si es posible obtener un valor sumando los elementos de un arreglo crece exponencialmente.

Ejercicio opcional: Fibonacci

n	Tiempo ejecución (ms)
0	0.0001430511474609375
1	1.3113021850585938e-05
2	5.2928924560546875e-05
3	2.4080276489257812e-05
4	8.106231689453125e-06
5	8.821487426757812e-06
6	0.0002906322479248047
7	1.1920928955078125e-05
8	1.5020370483398438e-05
9	2.193450927734375e-05
10	7.772445678710938e-05
11	6.914138793945312e-05
12	5.626678466796875e-05
13	8.392333984375e-05
14	0.00013303756713867188
15	0.000514984130859375
16	0.0003180503845214844
17	0.0005178451538085938
18	0.0010271072387695312
19	0.002184152603149414
20	0.0021440982818603516
21	0.0034902095794677734
22	0.005541086196899414
23	0.009152889251708984
24	0.01501011848449707
25	0.02458500862121582
26	0.03996992111206055
27	0.07444620132446289
28	0.1077580451965332
29	0.17058706283569336
30	0.26885104179382324

Gráfica de complejidad experimental:



Complejidad teórica:

$$T(n) = T(n - 1) + T(n - 2) + c$$

$$T(n) = -c + c_1 F_n + c_2 L_n$$

Notación O:

$$O(2^n)$$

Análisis:

La complejidad teórica del algoritmo para calcular la sucesión de Fibonacci es exponencial, y esta concuerda con la obtenida en la toma de datos experimental. Esto se debe a que en la gráfica anterior podemos ver que crece exponencialmente, al igual que en la ecuación.