

Taller 4 : Complejidad en algoritmos recursivos.

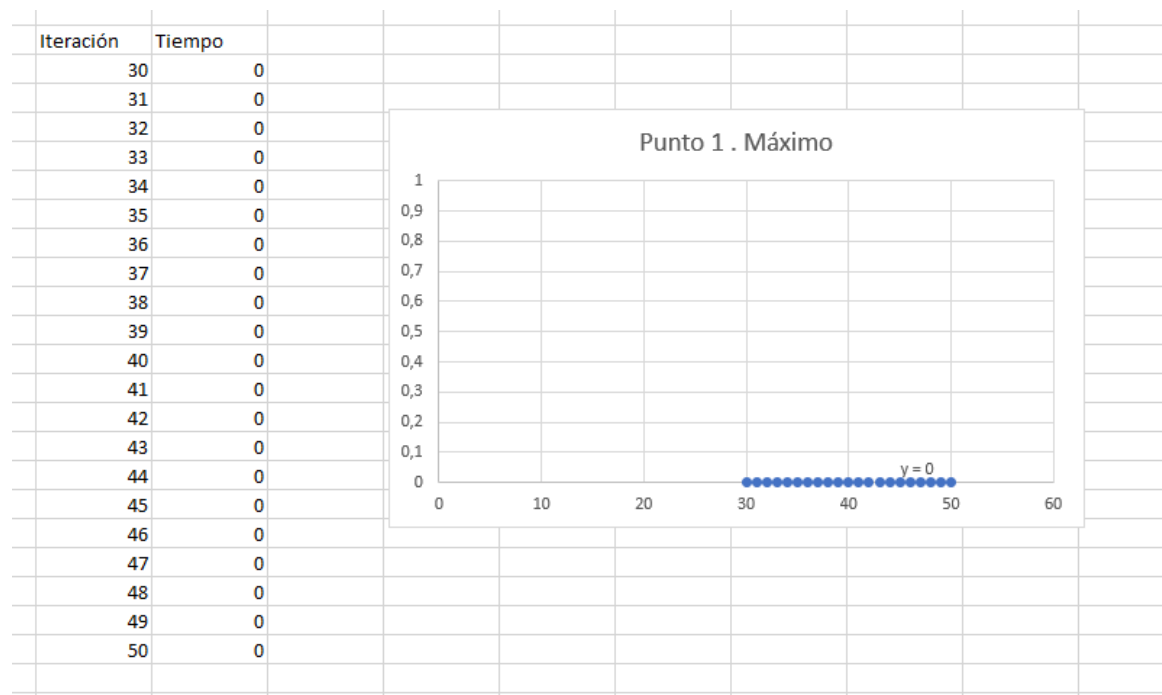
Stiven Yepes y Sara Rodriguez

Punto 1. Máximo de un arreglo.

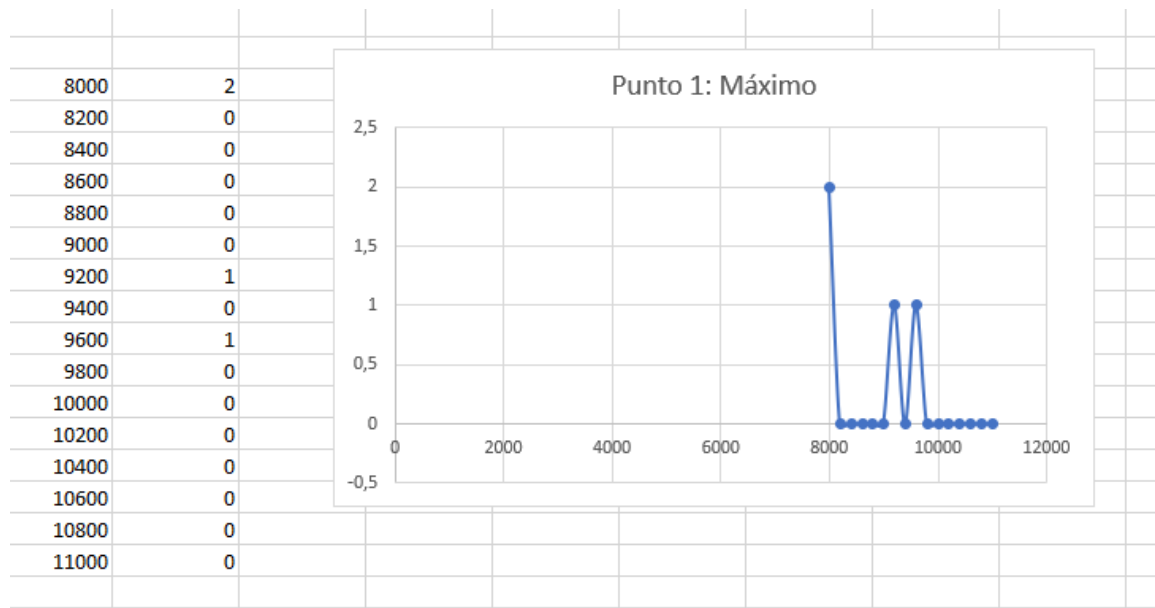
$$T(n) = \begin{cases} c_3 & n=0 \\ \end{cases}$$

$$\begin{cases} T(n-1) + c_2 \end{cases} \Rightarrow T(n) = c_2 n + c_1 \text{ (where } c_1 \text{ is an arbitrary parameter)}$$

Como se ve en el gráfico la complejidad es lineal, con una pendiente bastante baja, que se acerca mucho a cero, a menos que se usen valores bastante grandes. Esto tiene bastante sentido al compararlo con el resultado teórico, que no presenta potencias en ninguno de sus términos.



Al subir las iteraciones al máximo posible sin que se desborde la variable tenemos, nuevamente, una gráfica bastante similar, solo que con un par de desviaciones:



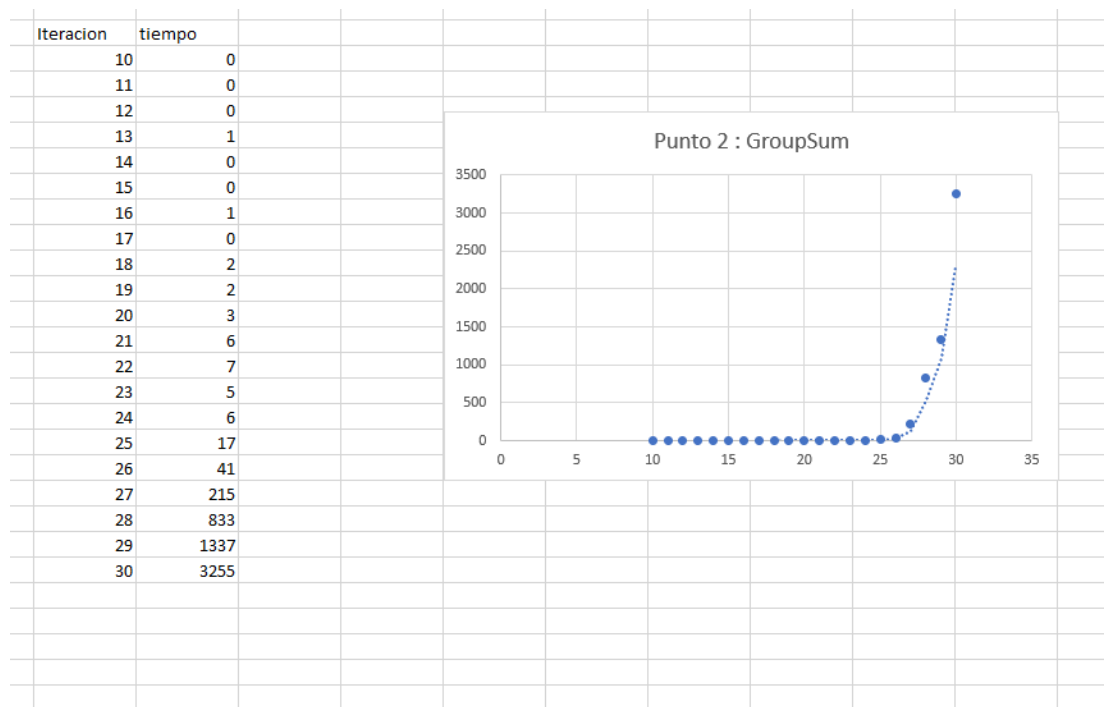
Como se puede observar el valor predominante sigue siendo cero.

Punto 2. GroupSum

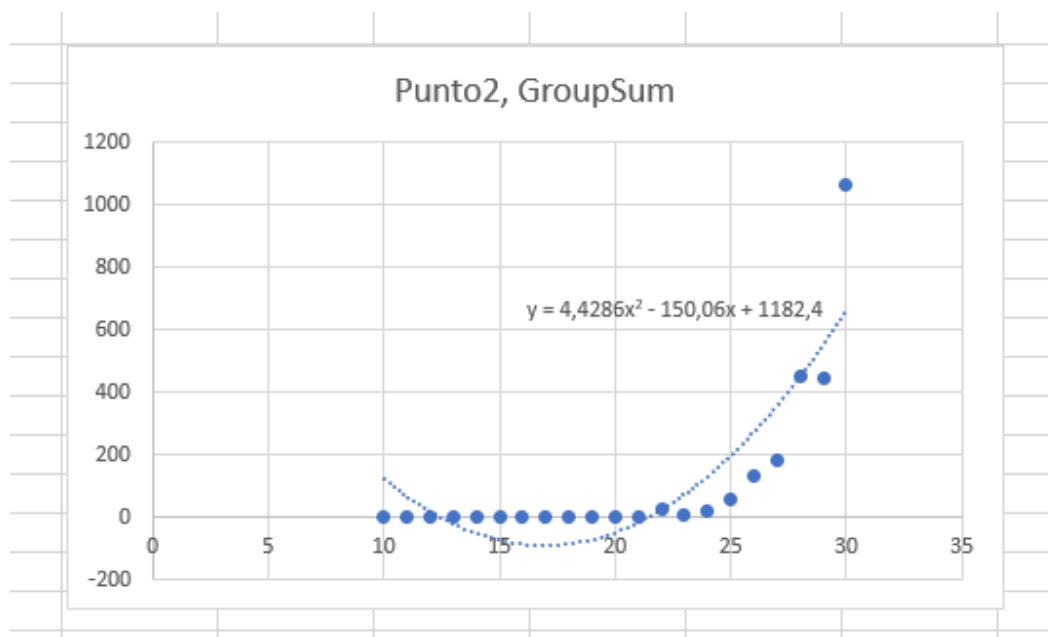
$T(n) = |c_2 \text{ start} > \text{nums.length} |$

$| T(n-1) + T(n-1) + c2 \quad | \Rightarrow T(n) = c_1 2^{(n-1)} + c_2 (2^n - 1)$ (where c_1 is an arbitrary parameter)

Como se ve en el gráfico se habla de una complejidad ascendente semejante a una función exponencial, sin embargo, los datos recolectados aún presentan mucha dispersión para ajustarse a una ecuación exponencial en Excel, por lo cual se especula que factores de procesamiento pueden intervenir en este aspecto como los programas que corren en segundo plano del computador donde se hizo el cálculo, entre otros. Esto tiene sentido al ser comparado con el modelo teórico en el cual se ve una complejidad en términos exponenciales.



La ecuación más cercana que pudo encontrar el solver es:



En la cual se confirma un comportamiento principalmente exponencial, aunque con un ruido o desviación notable.

Punto 3. Fibonacci

$$T(n) = \begin{cases} c_1 & n \leq 0 \end{cases}$$

$$\begin{cases} T(n-1) + T(n-1) + c_2 \end{cases} \Rightarrow T(n) = c_1 2^{(n-1)} + c_2 (2^n - 1) \text{ (where } c_1 \text{ is an arbitrary parameter)}$$

En esta ecuación se ve claramente que la función resultado es una función exponencial, lo cual coincide a la perfección con el modelo teórico que se tenía de antemano y se puede representar con una función exacta que describe la complejidad del algoritmo, y es: $y = 3E-06e^{0,4774x}$

