

Medición experimental de tiempos de ejecución en programas

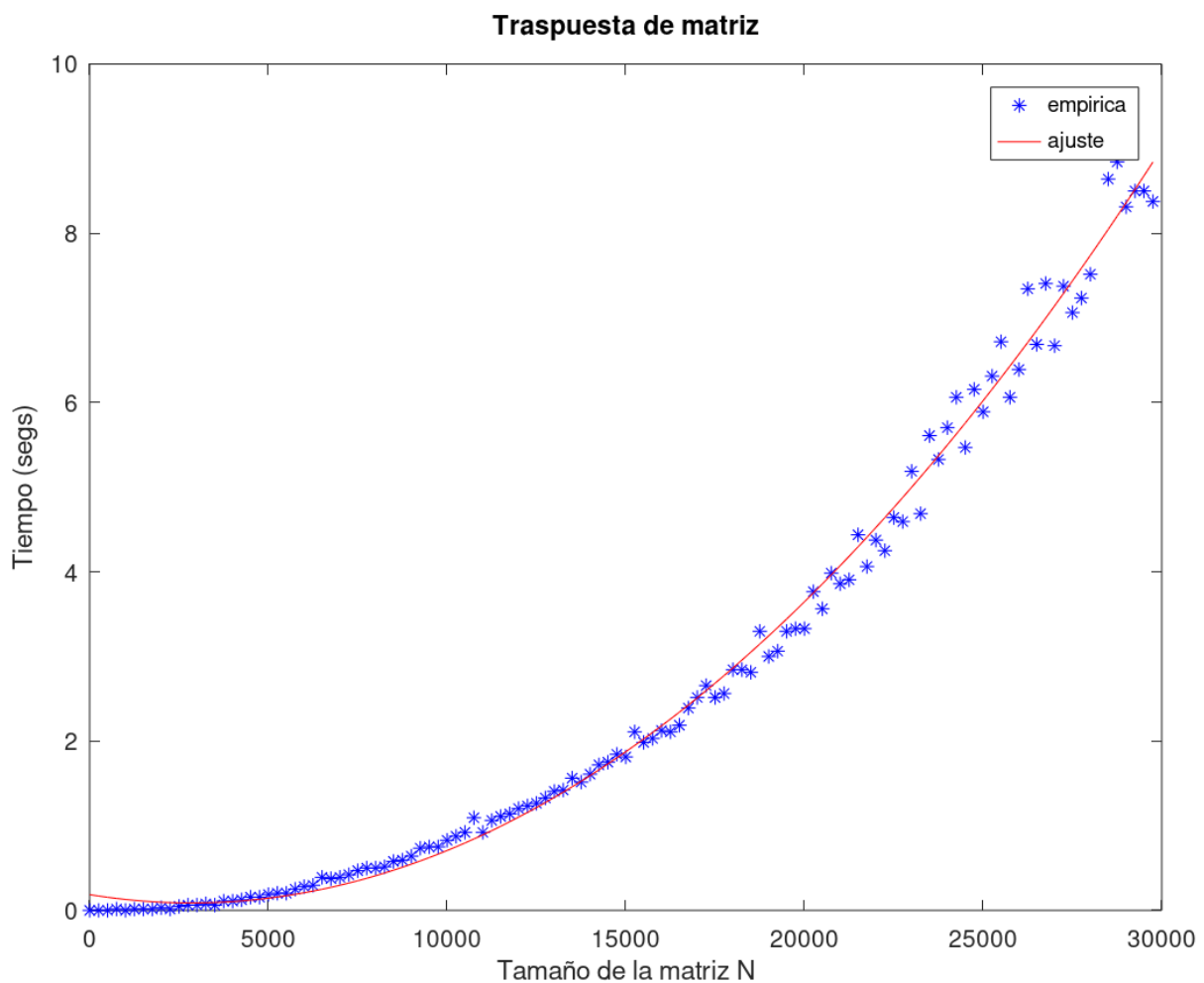
Aldán Creo Mariño <aldan.creo@rai.usc.es>

Introducción

Este informe tratará el contenido de la práctica 3. En ella, se hace una introducción a los tiempos de ejecución de código, midiendo el de dos funciones (`trasp` y `mult`). Analizaremos sus complejidades dependiendo del tamaño de matriz y extraeremos conclusiones en base a ello.

Antes de crear las funciones, pese a todo, fue necesario realizar un cambio en la función `crear`, cambiando el tipo de dato de `i` de `short` a `long`, ya que posteriormente se hace un bucle `for` que sobrepasa el rango posible de un `short` (ya que estamos operando con matrices de orden hasta 30.000).

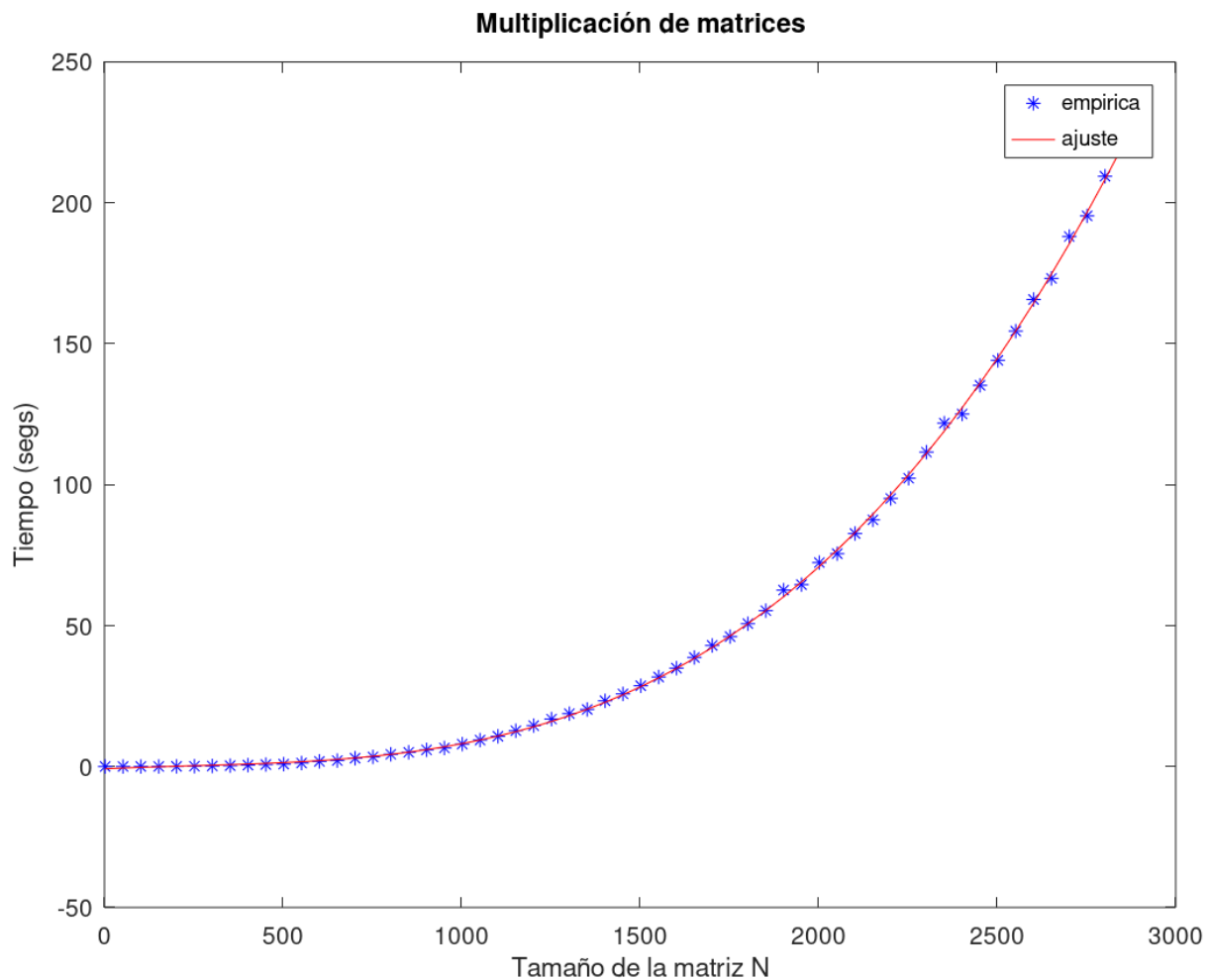
Función de trasposición



En la función `trasp`, el rango de la gráfica es de 0 a 30.000, ya que el TAD `matrizP` está diseñado para albergar tamaños de matriz de hasta 30.000 filas y columnas. La imagen, por su parte, va de 0 a aproximadamente 9 segundos. Este crecimiento es claramente más lento con respecto al de la función `mult`, que se recoge en el siguiente apartado.

El hecho de que esta función tenga complejidad cuadrática se explica por el uso de bucles `for` anidados. Dentro de un primer bucle que recorre todas las filas, tenemos otro bucle que recorre todas las columnas (en este caso, $\frac{1}{2}$ del número de filas). Es decir, se harán ($n_{\text{filas}} * \frac{1}{2} n_{\text{filas}} = \frac{1}{2} n_{\text{filas}}^2$) iteraciones (ya que el tamaño de la matriz es de $n * \frac{1}{2} n$). Y, como el coeficiente no se tiene en cuenta para calcular la complejidad, se cumple que $\text{tiempo_trasp} = O(n_{\text{filas}}^2)$. Por tanto, el tiempo de ejecución sigue una tendencia cuadrática.

Función de multiplicación



La gráfica de la función `mult`, va desde $n=0$ hasta $n=3.000$, aunque podría llegar a 30.000. El motivo es que los tiempos de ejecución se incrementan enormemente, mucho más rápidamente que en el caso anterior, lo cual hace inviable llegar a iterar con matrices de tamaños tan grandes como antes. En este caso, tenemos tres bucles `for` anidados, lo cual implica hacer el producto de dos elementos de `m1` y `m2`, respectivamente, $\frac{1}{2}n_{\text{filas_m1}}^3$ veces ($n_{\text{filas_m1}} * n_{\text{columnas_m2}} * n_{\text{columnas_m1}}$). De nuevo, podemos inferir que $\text{mult} = O(x^3)$.

Conclusiones

Venimos de ver que ambas funciones presentan una alta complejidad temporal, pero que la de la función de multiplicación es mucho peor. Este informe, pese a todo, no analiza las complejidades espaciales de ninguna de ellas, ya que no es el enfoque de la práctica, pero es fácilmente deducible, *a priori*, que la segunda también sería mucho peor.

Con respecto a los tiempos de ejecución asumibles, si tomamos un tiempo "asumible" como un máximo de 10 segundos, la función de trasposición nunca llegaría a exceder (en este modelo de computadora) ese máximo. Pese a todo, dado que la función de multiplicación es mucho más compleja, excede los 10 segundos que habíamos marcado de máximo a tamaños considerablemente más pequeños: en torno a $n = 1.100$.