

Informe Práctica 3

Programación II

Pablo Díaz Viñambres

a 21 de abril de 2020

1. Introducción

En este documento mediremos experimentalmente los tiempos de ejecución de los algoritmos de trasposición de matrices y de multiplicación de dos matrices. Comprobaremos así que estos algoritmos son respectivamente de $\Theta(n^2)$ y $\Theta(n^3)$.

2. Metodología

Generaremos matrices aleatorias de tamaños crecientes y mediremos el tiempo necesario para efectuar la operación. Todas las pruebas están realizadas siguiendo dos parámetros que se pueden elegir por el usuario dentro del programa:

t_{total} : el tiempo total que queremos que dure el experimento
 I : el incremento de tamaño de las matrices tras cada operación

Ajustaremos estos parámetros para obtener gráficas interesantes.

3. Trasposición de matrices

A continuación se presentan 2 gráficas obtenidas con distintos parámetros para la trasposición de matrices:

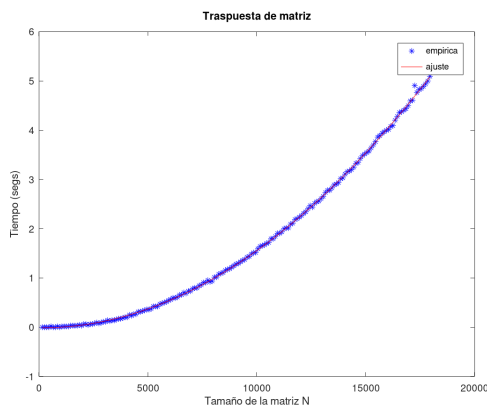


Figura 1: Trasp con $t_{total} = 300s$, $I = 100$
El rango resulta ser aproximadamente $[0, 20000]$

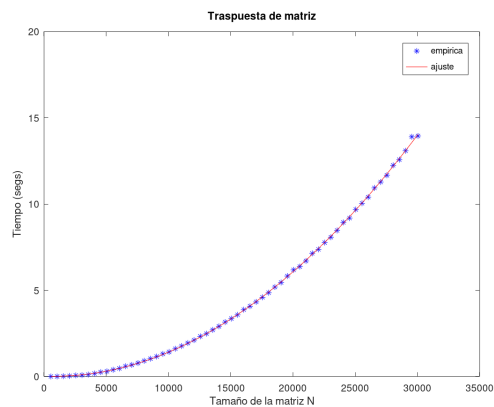


Figura 2: Trasp con $t_{total} = 300s$, $I = 750$
El rango resulta ser aproximadamente $[0, 31000]$

En ambas gráficas podemos observar el carácter cuadrático de la función. En la primera, observamos una mayor densidad de puntos al tener un menor incremento. En la segunda gráfica tenemos una versión más alejada, con un rango mayor y una cantidad de puntos menor debido a un incremento de tamaño mayor. Vemos que a partir de matrices de tamaño 25000 el tiempo de ejecución ya es de 10 segundos y deja de ser realmente práctico. Por la forma de la curva, podemos confirmar la hipótesis de que $TraspMatriz \in \Theta(n^2)$.

4. Multiplicación de matrices

A continuación se presentan 2 gráficas obtenidas con distintos parámetros para la multiplicación de matrices:

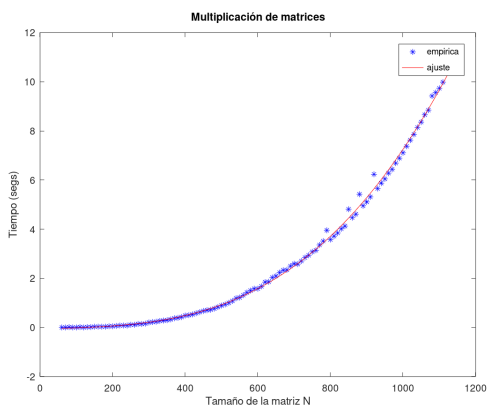


Figura 3: Trasp con $t_{total} = 300s$, $I = 10$
El rango resulta ser aproximadamente $[0, 1200]$

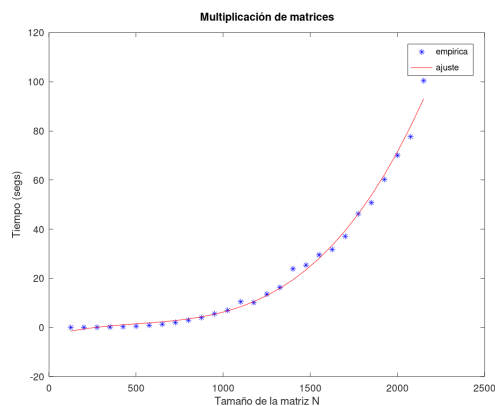


Figura 4: Trasp con $t_{total} = 600s$, $I = 75$
El rango resulta ser aproximadamente $[0, 2200]$

En este caso, podemos observar que el crecimiento es cúbico en vez de cuadrático. Esto nos lleva a que tengamos que usar un incremento menor al anterior para poder mantener un tiempo de experimentación razonable con una cantidad de puntos suficiente. Como consecuencia, los rangos son menores y se alcanzan tiempos de ejecución de 10 segundos ya con tamaños de 1100. Nuevamente, la segunda gráfica es una versión con menos zoom de la primera, y llegamos a ver con tamaños alrededor de 2000 tiempos de ejecución de 1 minuto. Todo esto indica a que este algoritmo es poco eficiente al trabajar con matrices grandes y rápidamente deja de ser práctico. Se confirma la hipótesis de que $MultMatrices \in \Theta(n^3)$.