

Análisis de algoritmos

DAA

Darío Domínguez González.....alu0101408095@ull.edu.es



Descripción de la arquitectura de ejecución	2
Resultados obtenidos	2
Representación gráfica	2
Conclusiones	3



Descripción de la arquitectura de ejecución

El programa se ejecutó en un entorno de desarrollo que utiliza un sistema operativo basado en Linux. La arquitectura del programa se basa en el patrón de diseño *estrategia*, donde se implementan dos enfoques diferentes para la multiplicación de matrices: uno que recorre las matrices por filas y otro por columnas.

Para evaluar el rendimiento de ambos algoritmos, se utilizaron matrices generadas aleatoriamente, con tamaños variables, para observar cómo se comportan los algoritmos frente a diferentes dimensiones de matrices. Los tiempos de ejecución se midieron utilizando la librería `<chrono>` de C++ para obtener resultados precisos en milisegundos.

Resultados obtenidos

A continuación, se presenta una tabla con los resultados obtenidos para distintos tamaños de matrices multiplicadas:

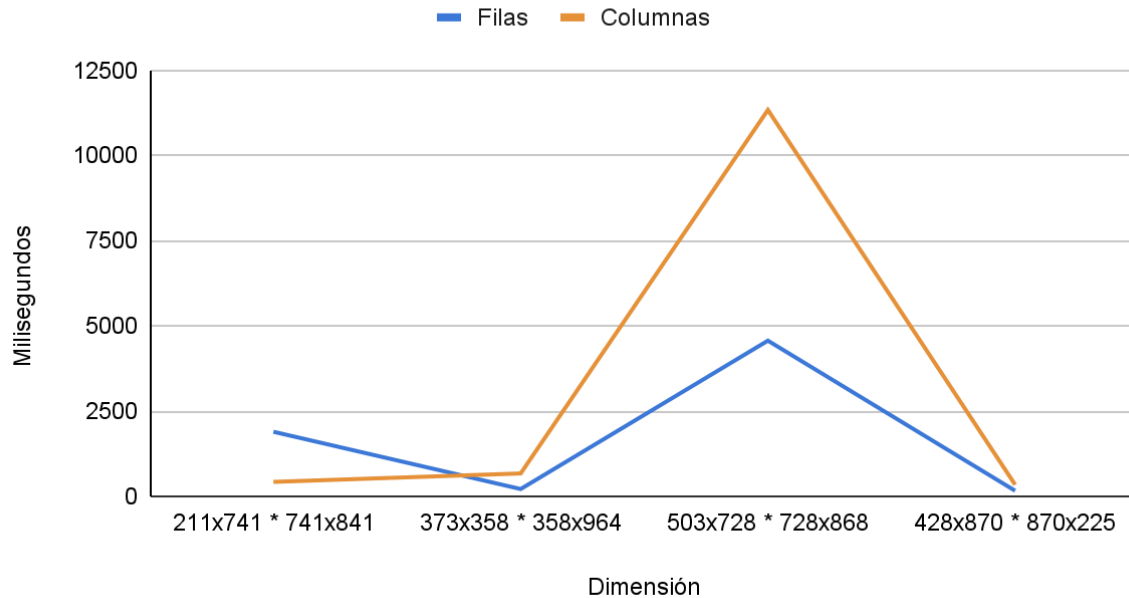
Tamaño de la matriz (filas x columnas)	Tiempo de ejecución (Recorrido por filas)	Tiempo de ejecución (Recorrido por columnas)
211x741 * 741x841	1894.17 ms	423.846 ms
373x358 * 358x964	213.913 ms	671.269 ms
503x728 * 728x868	4566.91 ms	11336.2 ms

Representación gráfica

La siguiente gráfica ilustra el comportamiento de los algoritmos de multiplicación de matrices para ambos enfoques (recorrido por filas y recorrido por columnas) a medida que aumenta el tamaño de las matrices:



Puntuación obtenida



Conclusiones

1. **Desempeño del algoritmo por filas:** El algoritmo de recorrido por filas resulta ser más eficiente en términos de tiempo de ejecución, especialmente a medida que el tamaño de las matrices aumenta. Esto puede deberse a que el acceso a los elementos en memoria se realiza de manera más secuencial, aprovechando mejor la caché.
2. **Desempeño del algoritmo por columnas:** El algoritmo de recorrido por columnas muestra tiempos de ejecución significativamente mayores en comparación con el de filas. Esto sugiere que el acceso a los elementos en las columnas no es tan eficiente, probablemente debido a la forma en que los datos están organizados en la memoria (es decir, las matrices están almacenadas en memoria en un formato de "fila mayor" en la mayoría de las implementaciones).
3. **Escalabilidad:** A medida que el tamaño de las matrices crece, los tiempos de ejecución aumentan considerablemente. En particular, el recorrido por columnas no parece escalar tan bien, lo que podría hacer que este enfoque sea menos adecuado para matrices muy grandes.
4. **Recomendación:** Para matrices grandes, se recomienda utilizar el algoritmo que recorre las matrices por filas, ya que ofrece un mejor desempeño computacional en comparación con el recorrido por columnas.