Realizar un algoritmo que resuelva la expresión (AB + CAT + BTD) donde A,B, C y D son matrices de NxN. Realizar una comparación entre una versión optimizada y una no optimizada (según el ejercicio de la 1º práctica).

Se deben completar brevemente las siguientes tablas.

Describir brevemente cuáles fueron las optimizaciones aplicadas.

|  |
| --- |
| Las optimizaciones que utilizamos son: ordenar las matrices por columnas y filas según sea conveniente para aprovechar la caché en las multiplicaciones. Eliminamos los llamados a funciones que accedían a los datos de las matrices ya que ocupaban más tiempo y memoria en accederlas y guardar sus parámetros. El resultado fue calculado y guardado dentro de una matriz previamente cargada y utilizada para optimizar el uso de la memoria. En el algoritmo de la multiplicación se cambió un único for donde se hacía todo el cálculo sobre las tres matrices por tres for, cada uno realizando una multiplicación de manera de aprovechar mejor la caché. |

Tiempos:

|  |  |  |
| --- | --- | --- |
| **N** | **No Optimizado** | **Optimizado** |
| **512** | 10,6619 s | 2,1746 s |
| **1024** | 152,0474 s | 17,4415 s |

Indicar brevemente. ¿Cuál es el algoritmo más rápido y por qué?

|  |
| --- |
| El algoritmo más rápido es el optimizado por lo expuesto en el punto anterior. Las optimizaciones del uso de la cache permiten acceder menos a memoria y por ende perder menos tiempo y aprovechar la localidad temporal y espacial de los datos. |

Describir brevemente las características de la arquitectura utilizada. (Procesador, memoria, niveles y tamaño de caché)

|  |  |
| --- | --- |
| CPU | Intel Core i5-2310 CPU @ 2.90GHz |
| Memoria | 8 GB |
| Caché | 4 x 32KB L1 Datos  4 x 32KB L1 Instrucciones  4 x 256KB L2 Unificado  6144KB L3 Compartida |