Trabajo Nro. 1

Optimización de algoritmos secuenciales

Pautas:

La entrega es en grupos de a los sumo dos personas.

Se debe entregar el código del ejercicio 1 optimizado.

Se deben completar brevemente las siguientes tablas según el ejercicio 3, relacionado a la multiplicación de matices utilizando la técnica por bloques.

Describir brevemente cómo funciona el algoritmo multBloques.c

El algoritmo se piensa de la siguiente forma: se divide cada una de las matriz en **N** partes o bloques.
La matriz A se divide en N partes: A_1 … A_N
La matriz B se divide en N partes: B_1 … B_N
La matriz C se divide en N partes: C_1 … C_N
 	
Tomamos el primer bloque A_1 de la matriz A, y recorremos el primer bloque B_1 de la matriz B.
Por cada fila del bloque A_1 , recorremos **r** columnas del bloque B_1 de la matriz B.
Por cada fila de A_1 y por cada columna de B_1, vamos actualizamos cada elemento C_1 de C
 	
Los siguientes for's son quienes realizan la verdadera multiplicacion (por bloque)
 	
for (i=0;i<r;i++){
for (j=0;j<r;j++){
desp = despC + i*r+j;
for (k=0;k<r;k++){
C[desp] += A[despA + i*r+k]*B[despB + k*r+j];
};
}
};
 	
Los siguientes for's son quienes realizan el movimiento por bloque

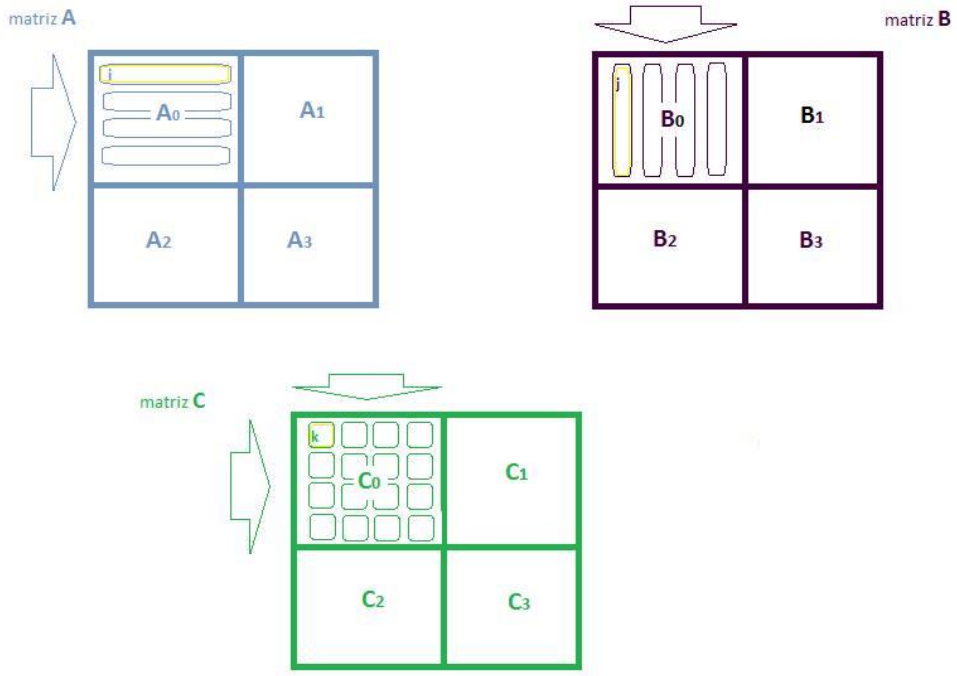
```
for (I=0;I<N;I++){
  for (J=0;J<N;J++){
    despC = (I*N+J)*sizeBlock;
    for (K=0;K<N;K++){
      despA = (I*N+K)*sizeBlock;
      despB = (K*N+J)*sizeBlock;
```

Para el ejemplo del dibujo, se interpreta de la siguiente forma:

Se multiplican los bloques A_o * B_p se actualiza en C_q , con o, p , q = 0, 1, 2, 3

Las multiplicaciones de los bloques se realizan en el siguiente orden:

A0 x B0 -> C0
A1 x B2 -> C0
A2 x B0 -> C2
A3 x B2 -> C2
A0 x B1 -> C1
A1 x B3 -> C1
A2 x B1 -> C3
A3 x B3 -> C3



Realizar la ejecución del algoritmo del ejercicio 1 optimizado y del ejercicio 3 para tres tamaños de bloque n1, n2 y n3 que deberán elegir.
Completar el siguiente cuadro comparativo entre ambos ejercicios con los tiempos de ejecución (en segundos) para los tamaños de matriz indicados.

N	Ejercicio 1	Ejercicio 3 (bloque tamaño n1)	Ejercicio3 (bloque tamaño n2)	Ejercicio3 (bloque tamaño n3)
512	0.852295	1.379848	0.895795	0.854130
1024	6.676549	17.580068	7.018649	6.507802

Las elecciones son las siguientes:
n1=2, n2=8, n3=32

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

El algoritmo por bloques del ejercicio 3 es en el mejor caso un poco más rápido que el del ejercicio1 (el algoritmo estándar para multiplicación de matrices). El motivo radica en que al elegir correctamente el valor del tamaño del bloque, podemos optimizar el uso de la memoria cache. Al aproximar **B** a **r** se logra el cálculo más óptimo

r: cantidad de bloques por lado de la matriz
B: tamaño de bloque

Indicar brevemente ¿De qué depende la elección del tamaño de bloque óptimo?

En un primer momento tendríamos que tener en cuenta que un gran influyente es el tamaño de la memoria cache. Si suponemos que tenemos una memoria cache de 3Mb y que solo se use para la ejecución del algoritmo, el número máximo del tamaño del bloque seria 724:

El tamaño de la memoria caché es 3Mb
Cada elemento double ocupa 2 bytes
Se tiene 3 bloques cargados en la cache (Ai, Bj, Ck haciendo referencia a la img)
3 Mb = 2 bytes * 3 * r^2
r = 724 aproximadamente

A partir de un **r** mayor a 724, por cada ciclo *for* implicaría leer dos veces de

memoria en vez de una vez. Por otro lado, se pudo apreciar que la mejor elección para el tamaño del bloque **b** es de valor igual o cercano a **r**. Por ejemplo para una matriz de 1024 elementos las mejores elecciones son las de r=32 y b=32

Describir brevemente las características de la arquitectura utilizada	
Core i3 3210m con l2 3MB 4Gb de RAM	