

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA**

**ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORAS**

**Laboratorio 3  
(Semestre 2023-0)**

**Indicaciones generales:**

- Debe enviar su solución, siguiendo el formato indicado en cada pregunta.
  - De no seguir estas indicaciones tendrá una penalidad en su puntaje.
  - **La hora de entrega es hasta culminada las 2 horas de evaluación de laboratorio.** Posterior a dicha hora, se anulará la evaluación.
  - Cualquier indicio de plagio resultará en la anulación de la prueba.
  - **La presentación, la ortografía y la gramática (incluyendo comentarios) influyen en la calificación.**
- 

**Multiplicación de matrices con números aleatorios (20 puntos)**

Desarrollar lo siguiente para evaluar el desempeño de los algoritmos de multiplicación matriz por matriz considerando las siguientes reglas:

- Las matrices A y B son cuadradas de  $N \times N$  y deben ser generadas con valores aleatorios enteros y positivos de máximo valor 9.
- Las entradas de los algoritmos deben ser la matriz A, la matriz B, una matriz llena de ceros que almacena el resultado y el tamaño de la matriz. En el caso de blocking, considerar el tamaño de bloque como entrada.

**(2.0 puntos) Reportar las características de su PC y las del V para realizar las comparaciones.**

Utilizando la plantilla **matriz\_matriz.py** y la plantilla **lib.c**, se le pide:

- a) (2.0 puntos) En el archivo **lib.c**, terminar de codificar la función *mat\_vec* la cual calcula el producto de dos matrices de  $N \times N$ .
- b) (2.0 puntos) En el archivo **lib.c**, terminar de codificar la función *mat\_vec\_block*, la cual calcula el producto de dos matrices de  $N \times N$  utilizando la técnica de blocking.
- c) (1.0 punto) Generar el archivo ejecutable .so tal como se indica en los comentarios del programa **lib.c**. Colocar un valor de  $N = 4$  y establecer el valor de la variable *max\_val\_mas\_1* en 10 (para cumplir las reglas indicadas líneas arriba en el programa) **matriz\_matriz.py**.

El ejecutar en el terminal `python3 matriz_matriz.py`, se deben generar 3 resultados iguales.

**Importante:** Al terminar estos incisos, llamar al jefe de laboratorio para que se le asigne el puntaje correspondiente de forma síncrona. Si no logra codificar la solución pasado la hora se le otorgará solo 3 puntos y se subirá la respuesta a paideia.

d) (5.0 puntos) Utilizando la plantilla **matriz\_matriz\_analisis.py**, realizar una gráfica tiempo\_ejecucion vs tamaño de matriz (N) para N = [32, 64, 128, 256] (variable ns). Considerar el tiempo de ejecución de la función codificada en a), b) y .dot de python. Generar gráficas distintas para los tamaños del bloque = [2, 4, 8, 16]. Responder en comentarios lo siguiente:

- ¿A partir de qué valor de N hay una mejora?
- ¿Todos los tamaños de bloque representan una mejora?

e) (5.0 puntos) Utilizando la plantilla **matriz\_matriz\_analisis.py**, realizar una gráfica tiempo\_ejecucion vs tamaño de matriz (N) para N = [128, 256, 1024, 2048] (variable ns). Considerar el tiempo de ejecución de la función codificada en a), b) y .dot de python. Generar gráficas distintas para los tamaños del bloque = [4, 8, 16, 32]. Responder en comentarios lo siguiente:

- ¿A partir de qué valor de N hay una mejora?
- ¿Todos los tamaños de bloque representan una mejora?

f) (3.0 puntos) Explicar las diferencias entre los algoritmos y justificar, considerando la localidad espacial y temporal y tomando en cuenta los tiempos de ejecución, cuál de los dos es el mejor y si la versión de Python es mucho más rápida en todos los casos.

**Importante:** Es probable que los tiempos de ejecución en el **inciso e)** tomen bastante tiempo, en dicho caso, se recomienda editar iteraciones del archivo **matriz\_matriz\_analisis.py**.

La presentación de su trabajo debe realizarse en un archivo PDF que tenga el formato L3\_codigopucp.PDF. Ese archivo y los códigos deben adjuntarse en un comprimido