

EXAMEN DEL PRIMER PARCIAL

Programación de Alto Rendimiento

4 de julio de 2025

Instrucciones:

A continuación se le presentan una serie de preguntas, léalas cuidadosamente, resuelvalos en forma clara y ordenada. Todas las respuestas deben de presentarse en formato .pdf y escritos en LaTeX; todo el código que presente debe se estar debidamente formateado usando el ambiente `minted`. No incluya archivos de código completos, puede solo presentar el código relevante a la solución a cada problema pero debe estar preparado para presentar el código completo y funcionando en caso de solicitarlo. Puede usar el código de este examen como plantilla.

1. Criba de Eratóstenes Mejorada:

Considere el archivo de la clase llamado `eratostenes/criba.c`, el cual calcula el número de primos menores que un número dado. El ejecutable `criba`, se puede usar de la siguiente manera:

```
$~ ./criba 8000000
El prime_count de 8000000 es 539777
The max Int is 2147483647
The max Long Int is 9223372036854775807
```

1.1. Inciso 1

Cuando se ejecuta `criba` para valores muy grandes, (por ejemplo, en mi computadora para valores mayores que 9 millones) el programa falla produciendo un *Segmentation Fault*. Interprete el error identificando las partes del código del programa que fallan. (10 pts)

1.2. Inciso 2

Las líneas del código 13 al 21 del programa `eratostenes/criba.c` son:

```

long int SieveOfEratosthenes(long int n)
{
    // Create a boolean array "prime[0..n]" and initialize
    // all entries it as true. A value in prime[i] will
    // finally be false if i is Not a prime, else true.
    bool prime[n + 1];
    // para inicializar un arreglo de datos
    memset(prime, true, sizeof(prime));

```

estas líneas inicializan el arreglo `prime` como un arreglo de tipo `bool`. Cambie el código del programa para inicializar el arreglo usando `malloc`. Recuerde liberar la memoria. (10 pts)

1.3. Inciso 3

Haga un análisis del efecto que tiene este cambio en el comportamiento del programa. Asegúrese de incluir en su análisis el tiempo de ejecución, si mejora o empeora el problema del Inciso 1. (10 pts)

2. Regresión Online

Considere la situación en la que existe una relación lineal desconocida entre las entradas x_t y las salidas y_t del tipo $y = mx + b$. Los datos (x_t, y_t) son recibidos de forma sucesiva e indefinida, por ejemplo, si x_t es el precio de la leche y y_t el precio del queso. En esta situación no existe un número determinado de datos, por lo que no se puede hacer regresión lineal tradicional, la cual en este contexto es conocida como regresión *offline*. Una alternativa, es hacer *regresión online* en donde los datos se utilizan una sola vez para actualizar los parámetros y luego se descartan. A continuación se describe el algoritmo de regresión online de *Widrow-Hoff*:

- Inicialice los parámetros: $(m, b) = (0, 0)$.
- Escoja un learning rate positivo α .
- Para todo $t = 1, 2, \dots$:
 - Lea el t -ésimo dato (x_t, y_t) .
 - Estime usando los parámetros actuales $\hat{y}_t = mx_t + b$.
 - Actualice los parámetros usando la fórmulas:

$$\begin{aligned}
 m &:= m - \alpha(mx_t + b)x_t \\
 b &:= b - \alpha(mx_t + b)
 \end{aligned}$$

2.1. Inciso 1

Implemente el algoritmo de Widrow-Hoff usando los datos con las mismas características que en el código en `gradiente/regresion/reg_utils.c`, es decir, para todo $t = 0, 1, 2, \dots$, el valor $x_t \in [0, 1]$ tiene distribución de probabilidad uniforme y y_t con ruido normal con media cero y varianza 0.1. (20 pts)

2.2. Inciso 2

Modifique el algoritmo anterior de manera que al generar un punto de dato (x_t, y_t) se genere usando dos hilos (“threads”) diferentes. Para ser específico, x_t y el ruido de la segunda componente deben de ser generados en dos hilo paralelos. (20 pts)

3. GNU Parallel

En las siguientes preguntas se usa el comando `parallel`.

3.1. Inciso 1

Use el comando `parallel` para crear en paralelo archivos de tipo .CSV. Genere 50 archivos en total usando 5 trabajos (“jobs”) en paralelo. Cada archivo debe ser generado usando el ejecutable `gradiente/regresion/generate_write`. La respuesta solo es el comando que se ejecuta en la terminal, no incluya los archivos generados. (15 pts)

3.2. Inciso 2

Escriba un programa que lea uno de los archivos generados en el ejercicio anterior y que calcule el promedio de las **columnas** del archivo. Específicamente el promedio de la primera columna deberá ser aproximadamente 0.5 y similarmente para la segunda columna. (15 pts)