

1IEE14 - Laboratorio 12

Instrucciones para el laboratorio:

- Materiales permitidos: Wiki del curso, apuntes de clase, consultar foros, tutoriales o documentación de python online.
- Está prohibido el uso de cualquier modelo de lenguaje como ChatGPT o alguno similar. A cualquier alumno que se le detecte que ha consultado un modelo de lenguaje se le pondrá nota 0(cero) en el laboratorio.
- Usted debe subir a Paideia 1 solo archivo comprimido (.zip o .rar) con el nombre L12_CODIGOPUCP.zip o L12_CODIGOPUCP.rar. Este archivo comprimido debe tener archivos de python(extensión .py) para cada pregunta. No se aceptarán soluciones en Jupyter notebook.
- Se espera que el nombre de los archivos sea: pregunta1.py, pregunta2.py, etc. A no ser que la pregunta especifique un nombre para su(s) archivo(s).
- El horario máximo permitido para subir el archivo es a las 10:00:00 pm. Pasada esa hora, habrá una penalidad de 2 puntos por cada minuto extra que se demore en entregar su archivo.

Pregunta 1 (4 puntos)

Dado el polinomio:

$$f(x) = x + 2x^2 + 3x^3 + 4x^4 + \dots + 10000x^{10000}$$

Calcular $f(2023)$. Para ello:

- (1 punto) Escriba una función f cuyo parámetro de entrada sea X y retorne el valor de $f(x)$. Este cálculo debe hacerse en serie, es decir no use multiprocessing. Calcule el tiempo de ejecución para $f(2023)$
- (3 puntos) Escriba la versión en paralelo de f , de tal manera de que el cálculo de $f(x)$ sea paralelo en 4 procesos y el cálculo se haga más rápido. Calcule el Speed up respecto a la parte a)

Nota: Al final de su archivo en Python, debe verificar que el resultado de la parte a) sea igual a la parte b). Para ello agregue la siguiente línea al final de su programa:

```
assert resultado_serial == resultado_paralelo
```

Dónde `resultado_serial` y `resultado_paralelo` son las variables de su programa que contienen los resultados de las partes a) y b). Si los resultados son iguales, la sentencia `assert` permitirá que su programa termine exitosamente; caso contrario le generará un error, y eso será evidencia de que sus resultados no coinciden.

Pregunta 2 (4 puntos)

Usted entra con su laptop a una cafetería a la que solía ir con frecuencia y se da cuenta que le han cambiado la contraseña a su WiFi. Así que decide intentar adivinar la nueva contraseña del WiFi.

Usted escucha una conversación de la mesa del costado en la que pudo notar que la contraseña tiene 3 letras, y sabe en base a experiencias pasadas de que siempre las **dos primeras letras son vocales**. En base a esta información, se le pide:

a) (2 puntos) Descargue la plantilla `p2_lab10_plantilla.py`, en esta plantilla escriba una función que retorne la contraseña correcta. El método que debe seguir es el método de fuerza bruta: Va a iterar sobre todas las combinaciones posibles en base a la información que tiene (3 letras, y las 2 primeras son vocales). Calcule el tiempo de ejecución.

b) (2 puntos) Usando multiprocessing, escriba una función que paralelice el método de fuerza bruta usado en la parte a). Para ello, va a crear 5 procesos, donde cada proceso toma como parámetro de entrada una de las 5 vocales y la asume como la primera letra de la contraseña.

Pregunta 3 (7 puntos)

Escriba un programa en Python que imprima si un número es primo o no. El algoritmo sugerido para saber si un número es primo es el siguiente:

Dado un número n , verificar si es divisible entre

Si es divisible entre algún número dentro de ese intervalo, no es primo; caso contrario sí es primo.

a) (2 puntos) Escriba una función que reciba como parámetro de entrada el argumento n y retorne True si el número es primo, o False en caso contrario (Implementación serial)

Imprima el tiempo de ejecución para $n = 2\ 345\ 678\ 911\ 111\ 111$

b) (3 puntos) Escriba una función divida la tarea de la parte en a) en 2 subtareas y permita paralelizar el cálculo de si un número es primo o no. Use 2 procesos.

Imprima el tiempo de ejecución de esta función para el mismo valor de n de la parte a). Calcule el Speedup. Use la función `assert()` para verificar que el resultado en la parte a) sea igual a la parte b)

c) (2 puntos) Escriba una función que reciba como parámetro un número x y que encuentre el primer número primo mayor a x . Para ello, su función debe tener un bucle, y en cada iteración calcular si es que los 2 números siguientes impares son primos o no usando multiprocessing con 2 procesos, 1 proceso por cada número impar a comprobar. (Solo se deben revisar los siguientes impares porque los pares ya se sabe que no son primos).

Por ejemplo, si $X = 24$:

Primera iteración: Se chequea si 25 y 27 son primos. Como no lo son, continuamos iterando

Segunda iteración: Se chequear si 29 y 31 son primos. Como 29 es primo, aquí se detiene el bucle.

Para corroborar en cada iteración si es que alguno de sus números impares es primo o no puede usar su función de la parte a o de la parte b, eso es opcional. Lo importante en esta pregunta es usar multiprocessing por cada número impar a verificar en cada iteración.

Su función debe imprimir el texto: "El siguiente número primo encontrado es P " donde P es el número que Ud ha encontrado.