1IEE14 - Laboratorio 5 2025-1

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 usar Github CoPilot. 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 L5_CODIGOPUCP.zip o L5_CODIGOPUCP.rar. Este archivo comprimido debe tener archivos de python para cada pregunta.

- 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 (2 puntos)

Escriba un programa en Python que use dos hilos para calcular la suma de los números en una lista. El algoritmo debe ser el siguiente:

1. Crear una lista de 100 números aleatorios.
2. Crear dos hilos: El primero recibirá solo la primera mitad de la lista, y el otro hilo recibirá solo la segunda mitad.
3. Cada hilo calculará la suma de los elementos que ha recibido.
4. El programa principal espera que los dos hilos hayan terminado. Cuando terminen, sumará las sumas parciales para obtener la suma final.
5. Imprimir el resultado de la suma final.

Pregunta 2 (4 puntos)

Se desea descargar un conjunto de 29 imágenes, que se pueden encontrar en los siguientes enlaces:

http://raw.githubusercontent.com/SebastianMerino/Threading/main/images/01.png
http://raw.githubusercontent.com/SebastianMerino/Threading/main/images/02.png
…
http://raw.githubusercontent.com/SebastianMerino/Threading/main/images/29.png

Descargue el archivo de Paideia llamado pregunta2.py el cual ya contiene un programa en Python para descargar las 29 imágenes secuencialmente.

a) (0 puntos) Modifique el programa para medir el tiempo de ejecución al ejecutar el archivo pregunta2.py. Anote este tiempo porque le será útil para la parte b). Esta pregunta vale 0 puntos porque solo tiene que medir el tiempo.

b) (2 puntos) Haga una copia de la parte a, renómbrelo pregunta2b.py, y modifique el programa para descargar los archivos utilizando hilos. Genere un hilo por cada imagen a descargar. Mida el tiempo de ejecución y calcule el speed up con respecto al resultado obtenido en a).

c) (2 puntos) Haga una copia de la parte b, renómbrelo pregunta2c.py, y descargue los archivos utilizando solo 3 hilos: Un primer hilo que descargue las imágenes del 01 al 09, un segundo hilo que se encargue de descargar las imágenes del 10 al 19, y un tercer hilo que se encargue de las imágenes 20 al 29. Mida el tiempo de ejecución. ¿Obtuvo mejores o peores resultados que en b)? ¿Por qué?

Nota: Para responder las partes b y c incluya un PDF con capturas de pantalla que sustenten sus respuestas.

Pregunta 3 (4 puntos)

Se desea instalar unos dispositivos que miden variables metereológicas en distintas partes del país. Para que estos dispositivos envíen sus datos sensados a una central en Lima, se están considerando 3 posibles tecnologías: SMS, 3G, o satelital. Cada una de estas tecnologías tiene un tiempo de latencia, el cual se puede modelar usando las siguientes fórmulas:

$$\text{Latencia de SMS} = \text{Tiempo\_de\_procesamiento\_de\_envío} + \text{Tiempo\_de\_procesamiento\_de\_recepción}$$

Donde:

Tiempo de procesamiento de envío: Valor aleatorio entre [10ms, 100ms]

Tiempo de procesamiento de recepción: Valor aleatorio entre [10ms, 100ms]

$$\text{Latencia de 3G} = 2 \text{ x } (\text{Tiempo\_de\_ida\_y\_vuelta} + \text{Tiempo\_de\_procesamiento})$$

Donde:

Tiempo de ida y vuelta: Valor aleatorio entre [100ms, 300ms]

Tiempo de procesamiento de recepción: Valor aleatorio entre [10ms, 100ms]

$$\text{Latencia de satelital} = 2 \text{ x Retardo\_de\_propagación} + \text{Tiempo\_de\_procesamiento}$$

Donde:

Retardo de propagación: Valor aleatorio entre [500ms, 700ms]

Tiempo de procesamiento de recepción: Valor aleatorio entre [10ms, 100ms]

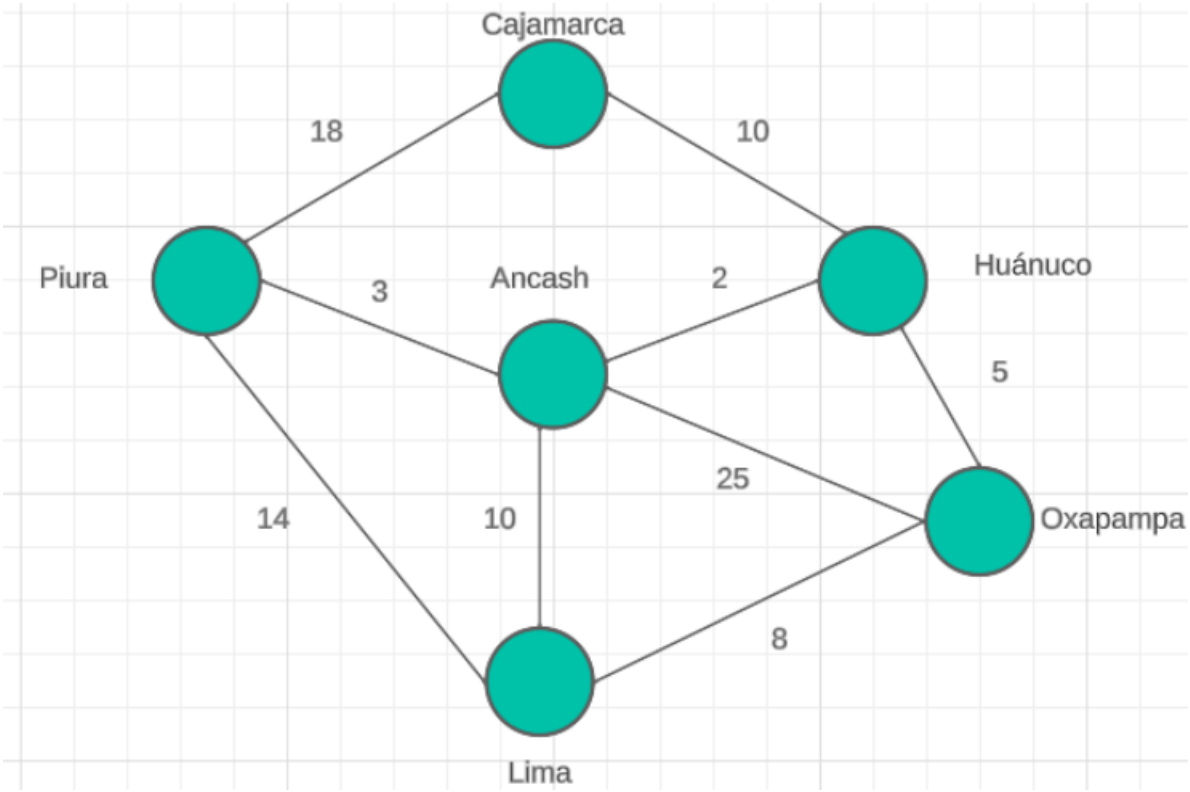
Su trabajo consiste en simular los tiempos de simulación de las 3 tecnologías usando corrutinas con asyncio. Para ello cree un archivo llamado pregunta3.py en el que escribirá un programa que ejecute 3 corrutinas de manera asíncrona. Cada una de las 3 corrutinas va a simular la latencia de las 3 tecnologías.

- Para generar un número aleatorio que simule la latencia, puede hacer uso de la función random.randint().
- Para simular la latencia en sí, deberá hacer uso de *asyncio.sleep()*. Esta función recibirá como parámetro de entrada el valor aleatorio que ha generado haciendo uso de random.randint(). De esta manera, cada rutina simulará la latencia de la tecnología. Una vez que el asyncio.sleep() termina de ejecutarse, la corrutina debe imprimir el siguiente mensaje:

  "La corrutina de la tecnología X tuvo una latencia de Y ms."
  Donde *X* puede ser "SMS", "3G", "Satelital" y *Y* es el resultado de la fórmula de latencia de cada tecnología.

Pregunta 4 (5 puntos):

Se tiene el siguiente gráfico, donde cada círculo representa una ciudad y las líneas representan las carreteras que unen cada ciudad. Los números en cada línea representan la cantidad de horas que tomaría viajar por tierra entre cada ciudad: Se desea averiguar cuál es el camino más corto para ir desde Lima (punto de partida) hasta Cajamarca (destino final).



Calcule el destino más corto usando el método de fuerza bruta: Ejecute una corrrutina por cada combinación posible y que cada corrtuina calcule el tiempo del recorrido.

Descargue la plantilla lab5_pregunta4_plantilla.py la cual contiene la lista de todas las combinaciones válidas posibles. Su programa debe comparar los tiempos de viaje y debe imprimir la ruta más corta.

Indicaciones de cómo se espera que implemente el programa:

- Debe usar corrutinas.
- Cada corrutina debe recibir como argumento de entrada la ruta que va a evaluar.

  El argumento de entrada debe ser una cadena de texto. Por ejemplo:

  - Si quiere que la corrutina evalúe la ruta Lima-Piura-Cajamarca, el argumento de entrada debe ser “LPC”.
  - Si quiere que la corrutina evalúe la ruta Lima-Oxapampa-Huánuco-Cajamarca, el argumento de entrada debe ser “LOHC”.
  - Dentro de cada corrutina, para simular la demora por cada tramo, debe usar la función asyncio.sleep(). Por ejemplo: Si su corrutina recibió como parámetro de entrada la cadena de texto “LPC”, su corrutina debe poder separar cada tramo, es decir, darse cuenta de que tiene que evaluar 2 rutas: Lima-Piura, Piura-Cajamarca. Como son 2 rutas que tiene que evaluar, ejecutará 2 veces el asyncio:

asyncio.sleep(14) # Esto es para simular la demora de Lima a Piura

asyncio.sleep(18) # Esto es para simular la demora de Piura a Cajamarca

Si lo desea, puede convertirlo a milisegundos para que no espere mucho tiempo.

- Las corrutinas deben retornar la cantidad de tiempo que demora recorrer su ruta. Por ejemplo, la que recibió como argumento de entrada “LPC” debe retornar 32, la que recibió como argumento “LAHC” debe retornar 22.

- Cuando todas las corrutinas hayan terminado de ejecutarse, su programa debe procesar los resultados e imprimir la siguiente frase:

  La ruta más corta es ABCD con una duración de X horas.