

Laboratorio 9 – Arquitectura de Computadoras

Nota: La pregunta 1 se desarrolla de manera síncrona y debe ser revisada por los jefes de laboratorio.

Para las preguntas 2 y 3, se desarrollarán de manera asíncrona y debe elaborar un informe en PDF mostrando los resultados obtenidos de sus implementaciones, caso contrario se le pondrá el puntaje de 0 en ambas preguntas. El plazo de entrega de las preguntas 2 y 3 es hasta el domingo 13/11 a las 10 p.m.

Pregunta 1 (10 puntos)

Se solicitará dos archivos: “client.py” y “server.py”.

- El servidor contará con 2 Threads:
 - **(2 puntos)** El primer Thread, luego de establecer conexión con el cliente, leerá un archivo csv.

Se tiene el archivo csv “[PartesDeElectrónica.csv](#)”:

La cabecera de este archivo es: [ID, Nombre, Número de parte, Cantidades, Peso, Costo unitario]

Este archivo contiene los datos de varios componentes electrónicos que se quieren comprar en un proveedor de China.

- **(2 puntos)** El segundo Thread, en un intervalo de 1 segundo, le enviará al cliente la cadena de caracteres de una fila (datos de un componente electrónico), así consecutivamente hasta completar de leer el archivo.
- El cliente, contará con 2 Threads en donde:
 - **(1 punto)** El primero se encargará de recibir la data del servidor.
 - **(3 puntos)** El segundo se encargará de procesar los datos de fila recibidos. Debe calcular lo siguiente:
 - Cálculo del costo total ($CT = \text{Precio unitario} * \text{cantidades}$)
 - Clasificación del componente por costo total:
 - (Menor a 25): Costo bajo
 - (25.0 – 49.9): Costo regular
 - (50 - 74.9): Costo alto
 - (75 a más): Costo elevado
 - Conteo de componentes que tengan costo elevado, conteo de componentes con cantidades peso mayor a 100g y con costo elevado, y conteo de componentes con costo bajo (conteos actualizados por cada componente que se procese).

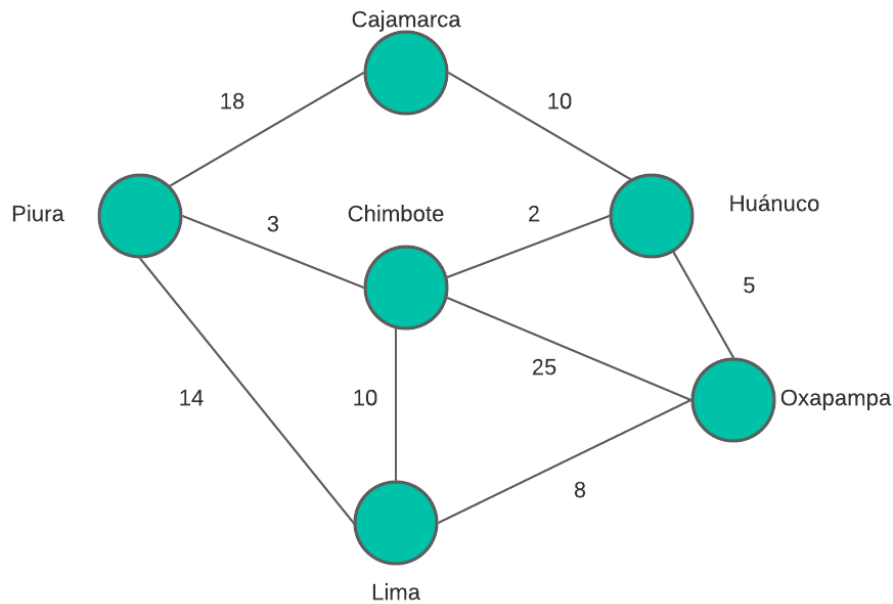
(2 puntos) Asimismo, el segundo thread imprimirá en terminal los resultados de cada fila de la siguiente forma:

```
-----Nombre: XXXX-----  
Costo total: XXX  
Clasificación por costo: XXXX  
Número de componentes con costo elevado: XXX  
Número de componentes con costo elevado y con peso mayor a 100g: XXX  
Número de componentes con costo bajo: XXX  
-----Nombre: XXXX-----  
Costo total: XXX  
Clasificación por costo: XXXX  
Número de componentes con costo elevado: XXX  
Número de componentes con costo elevado y con peso mayor a 100g: XXX  
Número de componentes con costo bajo: XXX  
-----Nombre: XXXX-----  
Costo total: XXX  
Clasificación por costo: XXXX  
Número de componentes con costo elevado: XXX  
Número de componentes con costo elevado y con peso mayor a 100g: XXX  
Número de componentes con costo bajo: XXX
```

Nota: solo debe imprimir en terminal si y solo si se tiene un nuevo dato a imprimir

Pregunta 2 (4 puntos)

Se tiene el siguiente gráfico, donde cada círculo representa una ciudad y las líneas representan las carreteras que unen a 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: cree un thread por cada combinación posible y que cada thread calcule el tiempo del recorrido.

Descargue la plantilla [pregunta2_lab9_plantilla.py](#) la cual contiene la lista de todas las combinaciones válidas posibles. Al final compare los tiempos de viaje para que su programa imprima la ruta más corta.

Pregunta 3 (6 puntos)

En la plantilla [pregunta3_lab9_622_plantilla.py](#) se ha definido la función `get_ntp_time()` la cual imprime la hora en un país determinado. Su argumento de entrada es cualquiera de las cadenas definidas en la lista `servidores_ntp[]`. Lo que hace internamente la función es obtener la hora de esas URL según el país solicitado. Por ejemplo:

Ejemplo1: si Ud. llama a la función de esta manera:
`get_ntp_time("1.es.pool.ntp.org")`; la función imprimirá la hora actual en España.

Ejemplo2: si Ud. llama a la función de esta manera:
`get_ntp_time("0.uk.pool.ntp.org")`; la función imprimirá la hora actual en Reino Unido.

Indicaciones sobre la función `get_ntp_time()`:

- Aparte de imprimir la fecha-hora, también retorna el timestamp en formato `datetime.datetime`.
- Los países disponibles en la plantilla son aquellos donde se encuentran las principales bolsas de valores del mundo donde uno puede comprar o vender acciones, lo cual será importante para este ejercicio.

No tiene que modificar `get_ntp_time()` por dentro, tampoco es necesario que entienda cómo es que funciona, solo úsela para lo siguiente:

a. **(3 puntos)** Escriba una función que itere sobre todas las cadenas en la lista `servidores_ntp[]` para determinar cuál es el país cuya bolsa de valores está más próxima a abrir con respecto a nosotros(en Perú). Asuma que todas las bolsas de valores abren a las 08:00am en sus respectivos países.

Por ejemplo: Supongamos que en actualmente en Perú fueran las 12:00, y en los otros países las horas son:

- UK: 17:00
- España: 18:00
- Estados Unidos: 12:00
- Hong Kong: 01:00
- Japón: 02:00

Entonces la respuesta correcta es que Japón sería el más próximo a abrir, porque es el que está más cercado a las 08:00am (pues solo le faltan 6 horas para abrir). Su programa al final **debe** imprimir el país que ha calculado tener la hora más cercana. Imprima el tiempo de ejecución de esta función.

Nota: Esos servidores son gratuitos y usan un tipo de protocolo que no responde el 100% de las veces, así que puede que a veces alguno de ellos no responda. Si Ud. ve que alguna URL se está demorando más de 10 segundos en imprimir su hora, cancele la ejecución de su programa(ctrl+C) y vuélvalo a ejecutar. Si el problema persiste, puede quitar esa URL de la lista servidores_ntp[].

- b. **(3 puntos)** Haga lo mismo que en la parte a) pero en vez de iterar sobre la lista, use threads para cada uno de los elementos en la lista servidores_ntp[]. Una vez terminado los threads, su programa **debe** imprimir el país que ha calculado tener la hora más cercana. Imprima el tiempo de ejecución de esta función.