Reto de programación.

Problema 1 (2 horas)

Pedro esta por irse de viaje, Pero antes necesita revisar el estado de sus calcetines. Cada calcetín tiene su propio color. Pedro quiere llevar el mayor número posible de pares de calcetines limpios, (los pares deben ser del mismo color).

Los calcetines están divididos en 2 cajones, sucios y limpios. Pedro únicamente tiene tiempo para un ciclo de lavado y su lavadora puede lavar un máximo de K calcetines. El quiere elegir los calcetines para lavar de tal forma que después de lavar el tenga el máximo número de pares limpios del mismo color. Es posible que algunos calcetines no encuentren su par, porque Pedro pudo haber perdido calcetines anteriormente.

Pedro tiene exactamente N calcetines limpios y M calcetines sucios, que están descritos en los arreglos L y S respectivamente. El color de los calcetines está representado por enteros (números iguales representan colores iguales).

Por ejemplo, si tuviéramos 4 calcetines limpios y 5 calcetines sucios:

Limpios

Sucios

Si la lavadora de Pedro puede limpiar hasta K = 2 calcetines, entonces el puede conseguir un máximo de 3 pares limpios de calcetines. Él puede lavar un calcetín rojo y un calcetín verde, indicados con el número 1 y 2 respectivamente. Entonces conseguirá un par rojo y uno verde de calcetines limpios.

Escriba una función solucion(K,L,S)

que dado un entero K (el máximo de calcetines que puede lavar la lavadora), dos arreglos L y S (conteniendo la representación de colores de N calcetines limpios y M calcetines sucios respectivamente), devuelva el máximo número de pares que Pedro puede llevar en su viaje.

Por ejemplo, dado K = 2, L= [1, 2, 1, 1] y S=[1, 4, 3, 2, 4] la función deberá devolver 3, como se explicó anteriormente

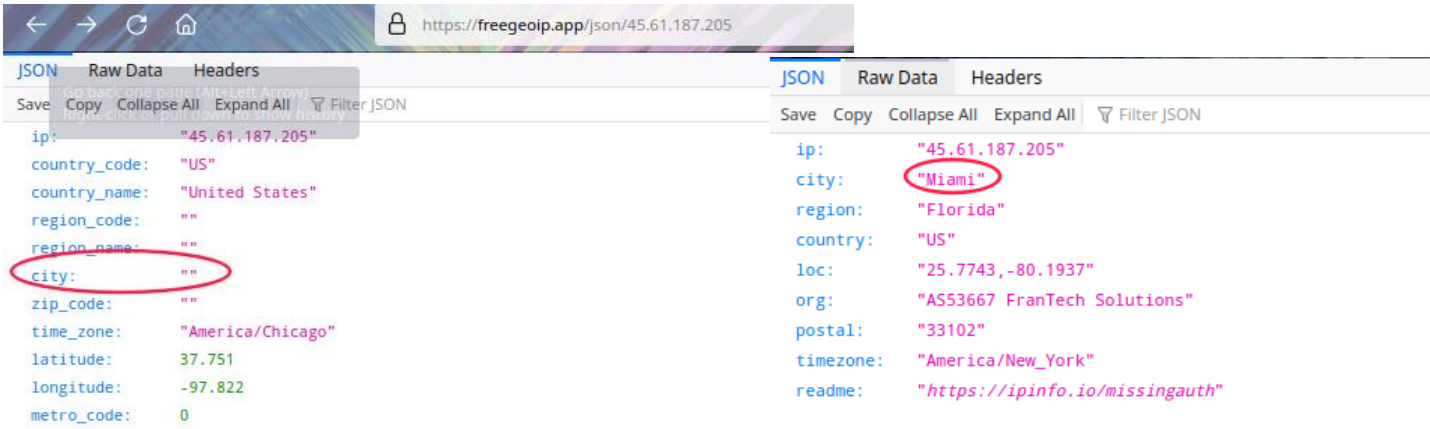
Asuma que

- K es un entero en el rango [0...50]
- cada elemento de los arreglos L y S es un entero en el rango [1..50]
- L y S no están vacíos y que cada uno contiene como máximo 50 elementos.

NOTA: Se espera que la solución no haga uso de librerías externas.

Problema 2 (2 días)

Existen diferentes bases de datos de acceso público donde se registra la geolocalización de de direcciones IP, sin embargo, no todas concuerdan en los datos que devuelven.



Se requiere determinar de forma automática la ciudad origen de una dirección IP después de consultar diferentes plataformas, en caso de que exista discrepancia en la información, indicar como "probable" la **ciudad origen que más veces coincidió en la consulta.**

Ejemplo de resultado:

Cuando las consultas **coinciden**:

Ciudad origen: Tokyo

Cuando las consultas **discrepan**:

Probable Ciudad Origen: Madrid

Se pueden hacer uso de las siguientes plataformas para consulta (al menos usar 2):

- https://geolocation-db.com/json/{DIRECCION_IP}&position=true
- https://ipinfo.io/{DIRECCION_IP}/json
- http://api.hostip.info/get_json.php?ip={DIRECCION_IP}&position=true
- https://json.geoiplookup.io/{DIRECCION_IP}
- https://freegeoip.app/json/{DIRECCION_IP}
- http://ip-api.com/json/{DIRECCION_IP}