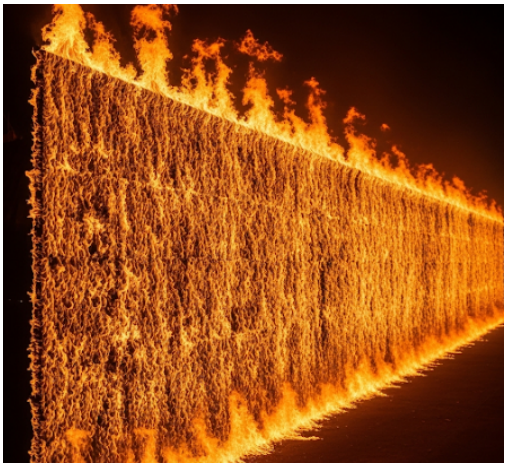
Paredes de fuego



Se sabe que en una red hay muchos paquetes dando vueltas, que pueden haberse enviado entre distintas computadoras. Cada paquete tiene una ip de origen, una ip de destino, un número de paquete, y los datos que se envían propiamente dichos. Ejemplo de paquetes dentro de una red:

A. Un paquete que va desde "192.168.1.102" a "192.168.1.103" con número 5 y datos "cómo".
B. Un paquete que va desde "10.1.1.55" a "192.168.1.102" con número 679 y datos "<h1> Proximo"
C. Un paquete que va desde "10.1.1.55" a "192.168.1.102" con número 676 y datos "
 genial "
D. Un paquete que va desde "192.168.1.102" a "192.168.1.103" con número 4 y datos "Hola che"
E. Un paquete que va desde "192.168.1.102" a "192.168.1.103" con número 6 y datos "estás?"
F. Un paquete que va desde "10.1.1.55" a "10.1.1.56" con número 7 y datos "gbmtp"
G. Un paquete que va desde"10.1.1.55" a "10.1.1.56" con número 7 y datos "gbmtp"

Para evitar distintos tipos de ataques informáticos y moderar la red, existen los firewalls, que son aparatos que regulan la red dejando pasar algunos paquetes y otros no. Cada firewall tiene una máscara de red y una serie de reglas, y sólo deja pasar los paquetes que le corresponden a su máscara de red y cumplen además todas las reglas. *Por ejemplo, un firewall con máscara "192.168" y sin reglas deja pasar todos los paquetes salvo el ejemplo F*. Es decir, los paquetes corresponden a una máscara de red si su origen o su destino comienzan con la máscara indicada.

Las reglas posibles para un firewall son las siguientes (pero podrían haber más):

a. **positivos**, que es una regla que sólo deja pasar los paquetes con número de paquete positivo.
b. **listaNegra**, que deja pasar los paquetes cuyo origen **no** está en una lista de ips dada. *Por ejemplo, la lista negra con 192.168.1.104 y 10.1.1.55 no deja pasar paquetes que vengan de esas ip y sí los que vienen de otras.*
c. **subMascara**, que dada una máscara de red sólo deja pasar los paquetes que corresponden a esa máscara. (es independiente de la máscara del firewall). *Por ejemplo, dejar pasar los paquetes de la submáscara "192.168.1"*.
d. **palabraClave**, que dada una palabra clave deja pasar los paquetes cuyos datos **no** incluyan esa palabra clave. *Por ejemplo, no dejar pasar los paquetes que contengan la palabra clave "apuesta".*
   Tip: En el repositorio se incluye una función que puede ayudar.

Se pide:

1. Modelar los paquetes, incluyendo los de ejemplo.
2. Modelar los firewalls y las reglas, sabiendo que se desea saber si un firewall **dejaPasar** un paquete.
3. Modelar los siguientes firewalls de ejemplo:
   a. Uno con máscara "10.1", con dos reglas de palabra clave (una prohibiendo los paquetes "apuesta" y otra prohibiendo los paquetes "xxx") y la regla positivos.
   b. Otro con máscara “192”, con una regla submáscara "192.168.1", con una regla de palabra clave "apuesta", y una lista negra con las ips 192.168.1.104 y 10.1.1.55.

4. Dado un origen, un destino, y una red (una lista de paquetes, que pueden ser de cualquier origen y destino), se pide:

    a. Conocer **queLeEnvio** el origen al destino en la red. Se espera una lista sólo con los paquetes cuyo origen y destino son los dados, ordenados por número de paquete.
    Por ejemplo, en la red que tiene todos los paquetes de ejemplo, si yo pregunto queLeEnvio "192.168.1.102" a "192.168.1.103", debo obtener 3 paquetes: el D, el A, y el E (ordenados así, por número de paquete).

    b. Escribir los tests automatizados para **queLeEnvio**.

    c. Saber si **estaCompleta** una comunicación en la red, para ese origen y destino. Eso ocurre cuando, en lo enviado de uno a otro, todos los números de paquete son consecutivos (cada paquete tiene el número anterior + 1). Resolverlo utilizando recursividad.
    Por ejemplo, para 192.168.1.102 a 192.168.1.103 en la red de ejemplo la comunicación está completa, porque están el paquete 4, el 5 y el 6, que son consecutivos. Pero para 10.1.1.55 a 192.168.1.102 la comunicación no está completa, porque está el 676 y el 679 pero no los del medio.

    d. Conocer el **mensaje** de un origen a un destino en la red. Se espera la concatenación de todos los datos de los paquetes correspondientes a lo que le envió el origen al destino.
    Por ejemplo, para 10.1.1.55 a 192.168.1.102 *el mensaje que hay en la red es* "
 genial <h1> Proximo".

    e. Conocer el **mensajeSeguro**, que es el mismo que el anterior pero pasando previamente los paquetes de la red a través de un firewall, con la salvedad de que si la comunicación no está completa luego de pasar por el firewall, debe devolver “Mensaje incompleto”.

Recordá avisar a tus docentes antes de irte. El docente cerrará el repositorio. Los repositorios no cerrados no se considerarán para la corrección.