



Proyecto #1

Fecha de entrega: semana 7

Viernes 08-nov-2024, 08:30 am

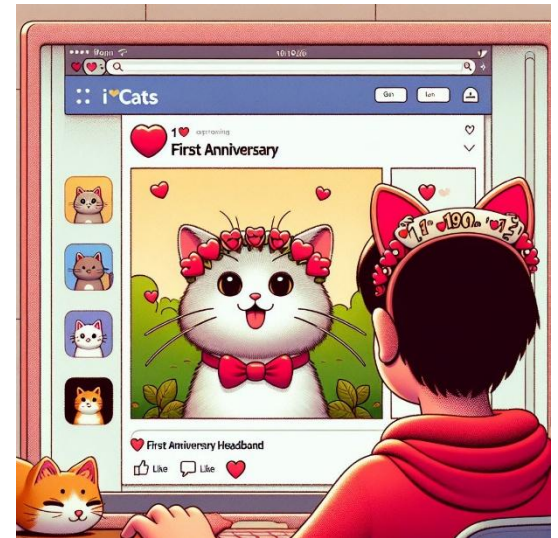
Valor: 35%

INTRODUCCIÓN / PLANTEAMIENTO

I ♥ CATS es una red social para fanáticos de gatos. Ha tenido mucho éxito durante su primer año. Un usuario publica fotos de gatos (una IA valida que solo suban fotos de gatos que cumplan ciertas condiciones), y TODOS los usuarios pueden verlas. Al momento de ver una foto, cualquier usuario (excepto el propietario) puede darle un único «*me gusta*».

Funcionamiento de **I ♥ CATS**:

- Esta red social evita que los usuarios conozcan su verdadera identidad, por lo que una vez completada la inscripción, se le otorga como identificador dentro de la red social un número entero mayor que cero, el cual se va generando en forma serial: 1, 2, 3, etc.
- En todo momento, un usuario de la red cuenta con dos listas (inicialmente vacías). La primera lista es «*candidatos a amigo*» y la segunda es «*amigos*».
- Cuando un usuario **B** le da «*me gusta*» a una foto del usuario **A**, se suceden las siguientes acciones:
 - El sistema chequea si **B** está en la lista de «*amigos*» de **A**. Si está, el sistema incrementa en uno el “contador de me gusta” entre **A** y **B**.
 - Si no, el sistema chequea si **B** está en la lista de «*candidatos a amigos*» de **A**. Si está, el sistema incrementa en uno el “contador de me gusta” entre **A** y **B**.
 - Si no, el sistema incluye a **B** en la lista de «*candidatos a amigos*» de **A**, y a **A** en la lista de «*candidatos a amigos*» de **B** y asigna uno al “contador de me gusta” entre **A** y **B**.
 - En todo momento, las listas son presentadas ordenadas de menor a mayor por el orden inducido por los identificadores de usuario.
- Puede darse un proceso de solicitud de amistad de **A** hacia **B**. Para esto, **A** y **B** deben estar en las listas de «*candidatos a amigo*» respectivas. Si el petitorio de amistad es aceptado por **B**, entonces el sistema saca a **A** y **B** de las listas de «*candidatos a amigo*» respectivas, y los transfiere automáticamente a las listas de «*amigos*» correspondientes. El “contador de me gusta” se transfiere con su valor al momento.
- Esta red social es muy sencilla:
 - Ningún usuario puede darse de baja de **I ♥ CATS**.
 - Una vez que dos usuarios son «*amigos*» no pueden “echar para atrás” la amistad.
 - Una vez que un usuario le da «*me gusta*» a una foto no se puede deshacer la acción.
 - Las acciones de los usuarios en la plataforma están perfectamente serializadas de manera que jamás se producen “irregularidades” (condiciones de carrera o abrazos mortales).





El «gestor de comunidades» de I ♥ CATS se ha reunido con usted, con una serie de necesidades:

1. Van a entregar un reconocimiento al usuario con más amigos (ojo: con ninguno o más de uno).
2. Van a entregar un “premio de consolación” al usuario con menos amigos (ojo: con ninguno o más de uno).
3. Quiere organizar fiestas por el primer aniversario para las «comunidades de amigos». Él le explica, que una «comunidad de amigos» es un grupo de usuarios en donde todo usuario en la comunidad tiene al menos un amigo en dicha comunidad y ningún amigo en otra comunidad. Tenga presente que una «comunidad de amigos» pudiera estar conformada por un único usuario aún “sin amigos”. Se quiere ordenar las comunidades de mayor a menor por su tamaño (número de usuarios que la integran). Dentro de cada «comunidad de amigos» se quiere hacer la clasificación de mayor a menor, de los usuarios que la integran, tomando como criterio el número de amigos, y si hubiere empate, por el orden de menor a mayor inducido por los identificadores de usuario.
4. Evalúa incorporar un cambio a la lista de «candidatos a amigo». Quiere agregar una información adicional, a la cual llama «grado de cercanía», para cada usuario en su lista de «candidatos a amigo» correspondiente. Sea C un usuario en la lista de «candidatos a amigo» del usuario A . El «grado de cercanía» entre A y C se define como: UNO (1) si existe un *fulano* amigo de C que es amigo de A , si no, como DOS (2) si existe un *mengano* amigo de C que es amigo de un *fulano* y *fulano* es amigo de A , y así sucesivamente. En caso de no existir una “cadena de amigos”, el «grado de cercanía» es INFINITO (∞). Así pues, ahora la lista de «candidatos a amigo» deberá presentar cada usuario que la integra, ordenada de menor a mayor por el «grado de cercanía», y si existe un empate, como segundo criterio de ordenamiento de menor a mayor por el orden inducido por los identificadores de usuario.

El «gestor de comunidades» de I ♥ CATS, le suministra dos archivos TXT:

Archivo: <i>candidatos.txt</i>	Archivo: <i>amigos.txt</i>
N	N
C	A
$[C_i C_j D]^*$	$[A_i A_j D]^*$

En donde por línea:

- N es el número de usuarios en I ♥ CATS.
- C es el número de relaciones de «candidato a amigo» y A es el número de relaciones de «amigo», las cuales se cuentan una sola vez, puesto que son simétricas.
- $[C_i C_j D]^*$ son tantas líneas como relaciones de «candidato a amigo» haya en un momento dado. C_i y C_j son los números de usuario separados con un espacio en blanco, con $C_i < C_j$. D es el “contador de me gusta”.
- $[A_i A_j D]^*$ son tantas líneas como relaciones de «amigo» haya en un momento dado. A_i y A_j son los números de usuario separados con un espacio en blanco, con $A_i < A_j$. D es el “contador de me gusta”.



Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
C12693 - Laboratorio de Algoritmos y Estructuras III
Trimestre septiembre-diciembre 2024

- El «**gestor de comunidades**» de I ❤️ **CATS** le asegura que los archivos están perfectamente generados, es decir, no existen inconsistencias de ninguna índole con los datos en los archivos TXT.



PETITORIO

1. Usted deberá entregar un informe en formato PDF tamaño carta, de máximo cuatro páginas. La primera página es una portada de identificación del curso, los estudiantes, etc. En las restantes páginas, deberá explicar de forma precisa, concisa y resumida, como resuelve la situación planteada, cuáles estructuras de datos utiliza, que algoritmos aplica, que limitaciones pudiera tener, etc. Este informe tendrá un valor de 10%.
2. Usted deberá programar una solución con Kotlin (**Main.kt**) haciendo uso de la librería **libGrafoKT**. El programa leerá los dos archivos TXT descritos: **candidatos.txt** y **amigos.txt**. El programa presentará por pantalla un informe que responda a las preguntas e inquietudes del «gestor de comunidades» de I ❤️ CATS. El formato de salida será el indicado más adelante. El programa y su ejecución tendrán un valor de 25%.

FORMATO DE SALIDA DE LA EJECUCIÓN DEL PROGRAMA

INFORME I ❤️ CATS

```
USUARIOS CON MAS AMIGOS=<CANTIDAD>
  1:<ID_USUARIO>:<CANTIDAD DE AMIGOS>:<LISTA DE AMIGOS>
  2:<ID_USUARIO>:<CANTIDAD DE AMIGOS>:<LISTA DE AMIGOS>
  ...
USUARIOS CON MENOS AMIGOS=<CANTIDAD>
  1:<ID_USUARIO>:<CANTIDAD DE AMIGOS>:<LISTA DE AMIGOS>
  2:<ID_USUARIO>:<CANTIDAD DE AMIGOS>:<LISTA DE AMIGOS>
  ...
COMUNIDADES DE AMIGOS=<CANTIDAD>
  COMUNIDAD 1
    1:<ID_USUARIO>:<CANTIDAD DE AMIGOS>:<LISTA DE AMIGOS>
    2:<ID_USUARIO>:<CANTIDAD DE AMIGOS>:<LISTA DE AMIGOS>
    ...
  COMUNIDAD 2
    1:<ID_USUARIO>:<CANTIDAD DE AMIGOS>:<LISTA DE AMIGOS>
    2:<ID_USUARIO>:<CANTIDAD DE AMIGOS>:<LISTA DE AMIGOS>
    ...
  ...
LISTA DE <<CANDIDATOS A AMIGOS>> POR USUARIO
  USUARIO 1
    1: <ID_USUARIO>:<GRADO DE CERCANIA>
    2: <ID_USUARIO>:<GRADO DE CERCANIA>
    ...
  USUARIO 2
    1: <ID_USUARIO>:<GRADO DE CERCANIA>
    2: <ID_USUARIO>:<GRADO DE CERCANIA>
    ...
```