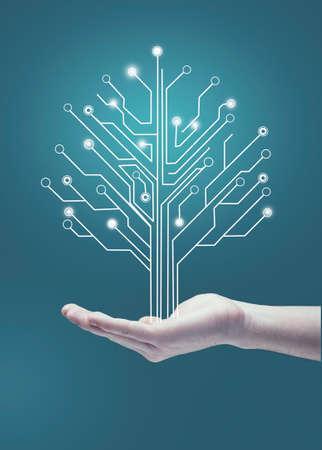
**Cómo hacer publicidad en una red social**

****

|  |  |
| --- | --- |
| **Nombre:** | Matías Cruces Gómez  Sebastian Rosas Urra  Mauro Muñoz Gonzales |
| **Fecha:** | 13/06/23 |
| **Tarea:** | Matemáticas discretas |
| **Profesor:** | Lilian Salinas |
| **Departamento:** | Depto. Ing. Informática y Cs. de la Computación |

**Introducción**

Abordaremos el problema de analizar una red social con el objetivo de proporcionar a los posibles anunciantes información relevante para tomar decisiones acertadas sobre qué usuario auspiciar en la plataforma. Para ello, se nos brinda información detallada sobre cada usuario, como su nombre, país de residencia, edad, si es creador de contenido o usuario regular, sus amigos y sus intereses.

La solución propuesta para abordar este problema consiste en desarrollar un programa en lenguaje C que permita cargar la información de la red social desde un archivo y generar un informe que cumpla con los requisitos solicitados. El programa solicitará al usuario el nombre del archivo que contiene los datos y procederá a procesar la información, filtrar la red social según los criterios del anunciante y finalmente identificar y contar las comunidades independientes.

La organización del informe será la siguiente: en primer lugar, se presentará una descripción detallada de la tarea y los datos disponibles en la red social. Luego, se explicará el método utilizado para generar el informe, incluyendo el proceso de filtrado de datos y la detección de comunidades independientes. Finalmente, se mostrará el informe generado, que incluirá el número de usuarios en cada comunidad, así como las listas de creadores de contenido y usuarios regulares que podrían tener mayor influencia dentro de cada comunidad.

**Objetivos:**

Analizar los datos de una red social y generar un informe que proporcione información útil a posibles anunciantes:

* **Comprender y procesar la información de una red social**: Extraer los detalles relevantes de cada usuario y establecer las conexiones entre ellos.
* **Filtrar la red social según criterios específicos**: Permite seleccionar un público objetivo de acuerdo con los criterios establecidos por el anunciante. Filtrando la red social en función del país de residencia, rango de edad e intereses de los usuarios.
* **Identificar y contar comunidades independientes**: Determinar y contar las comunidades independientes presentes en la red social filtrada. Analizando las conexiones entre los usuarios y detectar grupos de usuarios conectados directa o indirectamente a través de una sucesión de amistades.

### Modelo:

El modelo se genera a partir de una base de datos de usuarios almacenada en un archivo de formato .txt. Los datos son representados de la siguiente forma:

Texto

Descripción generada automáticamente

Por ejemplo 10 es el numero de usuarios, el primer usuario es Alejandro, seguido del país, su edad, su calidad de influencer (1=si, 0=no), su numero de amigos (3), los nombres de sus amigos, la cantidad de intereses que tiene y cuáles son.  
  
Con esta información se definió a la persona, con sus características, como un vértice en un grafo, y a la posible conexión con otras personas o vértices como la relación de amistad mutua entre ellos o la calidad de influencer de la persona de la cual se considera como amiga.

Para esto se definen dos estructuras principales Persona y Comunidad. La estructura de persona almacena información acerca de las características e intereses de cada persona, se almacena su nombre, país, edad, su calidad de influencer, su número de amigos y cuáles son, su número de intereses y cuáles son y también su calidad de ‘Visitado’. La estructura de Comunidad representa un grupo de individuos donde cada uno es un miembro de la comunidad y, en un principio, tienen en común la calidad de amigos entre ellos, aunque también se pueden usar filtros dentro del programa para indicar qué características específicas se buscan en estas comunidades. Las comunidades almacenan el número de miembros que contienen y el nombre de cada uno, se pueden generar varios arreglos de comunidades para diferenciar diferentes grupos de personas.

Por ejemplo, para el data set:  
  
10

Alejandro Chile 29 0

3 Pedro Juan Diego

2 Cocina VideoJuegos

Pedro Argentina 20 1

2 Alejandro Diego

1 VideoJuegos

Juan Chile 25 0

2 Alejandro Diego

1 Cocina

Diego Argentina 22 0

1 Juan

1 Cocina

Maria Bolivia 30 0

1 Juana

2 Cocina VideoJuegos

Juana Bolivia 31 1

1 Maria

1 VideoJuegos

Constanza Argentina 25 1

2 Pedro Marcela

1 Videojuegos

Marcela Chile 20 1

1 Pedro

2 VideoJuegos Cocina

Cecilia Chile 25 0

1 Mario

1 Cocina

Mario Chile 25 0

1 Cecilia

1 Cocina

Sin colocar filtros, el programa nos entrega:  
Texto

Descripción generada automáticamente

Lo cual representa el grafo:  
  
Diagrama

Descripción generada automáticamente  
  
  
Nota:   
El programa usa las librerías: stdlib y string, agregar las siguientes líneas de comando para compilar:  
#include <stdlib.h>  
#include <string.h>

### Implementación:

La idea fue de tratar de hacer funciones que cumplan ciertos objetivos y que se ejecuten por partes, desde el main.

El código comienza pidiendo al usuario el nombre del archivo de texto a leer, también ofrece la opción de aplicar filtros según la necesidad del cliente. Se pueden aplicar filtros para edad, país o intereses específicos. Una vez que se entrega la información el programa recorre el archivo de texto y crea arrays para los individuos y sus características y para las comunidades y sus miembros. Para el array de Persona se usa la variable ‘personas’ y para Comunidad se usa ‘comunidades’. Al principio las variables NumeroDePersonas y NumeroDeComunidades empiezas en 0, se lee el número total de personas y se le da un valor a NumeroDePersonas, luego se crean loops ese número de veces para poner los datos de cada persona en el array de ‘personas’.  
La función ‘AgregarPersona’ es la que crea una instancia de persona y la añade al array. La función ‘AgregarAmigos’ añade y asigna los amigos a una cierta persona. La función ‘AgregarIntereses’ añade y asigna los intereses a las personas.

El programa luego cierra el archivo de texto y analiza la información obtenida.

La función ‘FormarComunidades’ se encarga de formar comunidades de acuerdo a los datos obtenidos en el array creado ‘personas’. Esta función itera sobre la lista de todas las personas y verifica si han sido visitadas o no, si no han sido visitadas se hace uso de la función ‘AgregarMiembroAComunidad’ para agregar esa persona a la comunidad actual, luego a partir de esa persona se usa el algoritmo BFS la cual toma como punto de inicio el índice de esa persona y de su comunidad (startPersonIndex y indicecomunidad), el algoritmo recorre la lista de amigos de esa persona de acuerdo a las reglas de amistad en el enunciado, y las va agregando a la comunidad si corresponde con la ayuda de la función ‘FormarComunidades’. Una vez que se completa esa comunidad, se crea una nueva y se vuelve a repetir el procedimiento para las personas que no han sido visitadas.  
  
Con las comunidades ya formadas se empiezan a analizar de acuerdo a los requerimientos del usuario, se aplica la función ‘filtrarPersonas’ para aplicar los distintos tipos de filtros e imprimir en pantalla solo las personas que pasen por ellos.  
  
La mayor parte del main itera en un ciclo do el cual se va repitiendo indefinidamente según lo requiera el usuario.

**El problema**

Surgió un problema con el código BFS (que es donde suponemos que se encuentra), el código corre bien para el data set de ejemplo dado pero no se adecúa del todo bien a nuevos sets.

Puede haber sido un problema con la escritura de las leyes de amistad o con el ciclo BFS en sí, lo que complicó también que ejecutáramos la parte en donde se pedía los mejores usuarios y mejores influencers. El código no respondía bien y empezaba a formar comunidades erróneas. Por temas de tiempo decidimos dejarlo hasta este punto y tratar de analizarlo con más práctica.

**Conclusión**

Al principio fue siempre prueba y error, lo que nos complicaba era como ejecutarlo ya que teníamos varias ideas, pero al final vimos que una se adecuaba más a una implementación que era más comprensible para cada uno.

Tuvimos algunas complicaciones con la comunicación y traspaso de archivos al comienzo, se proponían diferentes ideas y puede que nos hayamos demorado mucho en la etapa de planificación y asignación de tareas, pero de a poco fuimos mejorando ese aspecto con comentarios en el código, uso de github, y traspaso de responsabilidades. Desde el punto de vista técnico, nos costó trabajar con el algoritmo BFS y con los filtros, que cumpla correctamente con las iteraciones y que interactúe con las demás funciones, ya que se generaba conflicto con los vértices visitados y los no visitados. Lo que aprendimos de ello fue que el orden del trabajo es fundamental, la correcta identación y nombramiento de variables.

A pesar de todo, nos dio el concepto de trabajar con puntos de datos con información relacionada y a como manejarlos de forma de poder interpretarlos por separado o en grupos con ciertas características. Se observó también como los conceptos de vértices, aristas y grafos en general son llevados a la práctica y otras posibilidades que se puedan modelar con ellos.