

Escenario de la propuesta del TRABAJO INTEGRADOR para el segundo Parcial: “Red Social Universitaria”

Objetivos de la Tarea

- Aplicar paradigmas de diseño y análisis de algoritmos en un entorno realista.
 - Desarrollar en un lenguaje de programación específico (Java) un sistema que modele aspectos clave de una red social.
 - Analizar la complejidad y el rendimiento de los algoritmos elegidos.
 - Promover la toma de decisiones fundamentadas sobre qué técnica algorítmica es la más apropiada para cada problema.
-

Consigna General

Una empresa de tecnología busca construir un prototipo de una red social universitaria. Los estudiantes, trabajando en grupos, deberán implementar en Java una solución para al menos **dos problemas principales** y un **desafío opcional**. El proyecto debe ser modular, y cada equipo deberá justificar sus decisiones de diseño en un informe y una presentación.

Problemas Principales (elegir al menos dos)

1. Gestión de Publicaciones: El Muro de Novedades

Los usuarios generan publicaciones que deben ser organizadas y presentadas en un muro. El sistema debe permitir dos vistas: una cronológica y otra por relevancia.

Desafío: La vista por relevancia pondera factores como el número de "me gusta" y la antigüedad. Para un gran volumen de publicaciones (millones), ¿cómo se puede garantizar que la lista de las más relevantes se mantenga actualizada y disponible de forma eficiente, sin reordenar todo el conjunto cada vez que se añade una nueva publicación? ¿Es posible mejorar el rendimiento al dividir el conjunto de datos en partes más pequeñas y procesarlas de forma independiente antes de unificarlas?

2. Asignación de Publicidad: Maximizando el Alcance

Cada usuario tiene un perfil y un tiempo máximo de exposición a anuncios. La empresa tiene una cartera de anuncios, cada uno con un costo y un valor de alcance potencial. Se necesita un algoritmo que asigne anuncios a usuarios de manera que se maximice el alcance total de la publicidad sin exceder el presupuesto.

Desafío: ¿Cómo se puede abordar este problema de optimización combinatoria de manera eficiente? Considera un enfoque que construya una solución óptima a partir de la solución óptima de subproblemas más pequeños, recordando la solución de cada uno para no calcularla dos veces.

3. Red de Amistades: Mínima Conectividad

La red social se modela como un grafo de usuarios conectados por relaciones de amistad. El objetivo es determinar el conjunto mínimo de conexiones necesarias para asegurar que todos los usuarios de un campus estén conectados entre sí. Esto es crucial para minimizar los costos de infraestructura o las dependencias.

Desafío: ¿Qué estrategia permite construir esta red de conexión, añadiendo las "conexiones" más baratas una por una, siempre que no se creen redundancias o bucles en el proceso?

4. **Recomendación de Amigos: El Vínculo más Corto**

El sistema debe sugerir amistades potenciales a un usuario dado. La recomendación se basará en la "distancia" en la red de amistades: se considerará amigo potencial a cualquier usuario que se encuentre a la "menor distancia" del usuario actual en la red. Las distancias pueden ser ponderadas por la cantidad de interacciones.

Desafío: Diseñar un algoritmo para encontrar la ruta de menor costo (o menor número de saltos) desde un usuario a todos los demás en la red. ¿Cómo se garantiza que, en cada paso de la construcción de la ruta, se está eligiendo el camino más directo? ¿Qué costo tiene ejecutar esta búsqueda en una red con miles de usuarios?

5. **Búsqueda de Comunidades: Con Restricciones**

La plataforma debe permitir a los usuarios buscar comunidades o grupos que cumplan con criterios específicos. El problema es encontrar todas las configuraciones posibles de grupos que cumplan ciertas condiciones de inclusión o exclusión de usuarios (ej: "debe incluir a A y B pero no a C").

Desafío: Considera una estrategia que descompone el problema en la exploración de un árbol de decisiones, donde en cada paso se toma una decisión sobre la inclusión de un usuario, y se puede podar o descartar ramas enteras que no cumplen con las condiciones de manera temprana.

Desafíos Opcionales (elegir al menos uno)

1. **Simulación de Bloqueos y Conexiones Alternativas**

Un usuario bloquea a otro, rompiendo una conexión en la red. El sistema debe ser capaz de determinar si el grafo de amigos sigue siendo completamente conexo. Si no lo es, ¿cómo se podría encontrar un conjunto mínimo de nuevas conexiones para restablecer la conectividad total?

Desafío: Piensa en cómo se puede modelar el problema como una búsqueda exhaustiva de combinaciones de conexiones, evaluando el impacto de cada adición o eliminación.

2. **Optimización de Interacciones en la Portada**

Cada publicación tiene un valor de beneficio (por ejemplo, likes + comentarios) y un tamaño. El espacio en la portada es limitado. El sistema debe seleccionar un subconjunto de publicaciones destacadas que maximice el beneficio total sin exceder el espacio disponible.

Desafío: ¿Cómo se puede resolver este problema que tiene la propiedad de que la solución óptima para el problema completo se puede construir a partir de las soluciones óptimas para subproblemas más pequeños?

3. Asignación de Administradores a Grupos

"La red social necesita asignar administradores a diferentes grupos de usuarios. Cada administrador tiene un valor de eficiencia diferente para cada grupo (por ejemplo, un administrador puede ser muy bueno para un grupo de programación pero menos para uno de arte). El desafío es encontrar una asignación que cubra a todos los grupos, de modo que la suma total de las ineficiencias sea la mínima posible."

Desafío: "Analiza si este problema puede ser resuelto construyendo la solución de forma incremental. Considera si, para encontrar la mejor asignación para los primeros N grupos, podrías apoyarte en la solución óptima ya encontrada para los $N-1$ grupos anteriores. ¿Cómo podrías almacenar y reutilizar los resultados de los cálculos intermedios para evitar la repetición de trabajo innecesario?"

4. Exploración de Rutas de Influencia

Determinar todas las posibles cadenas de influencia entre usuarios (ej: una publicación de A llega a D a través de B y C). Se busca la cadena más efectiva en términos de alcance o velocidad de propagación.

Desafío: Diseñar un algoritmo que recorra exhaustivamente la red, con la capacidad de detenerse al encontrar una solución óptima y evitar caminos que ya se sabe que no conducirán a un mejor resultado.

Entregables (12/2)

- **Código en Java:** Un proyecto modular, comentado y ejecutable, con casos de prueba para cada problema resuelto.
- **Informe Escrito:** Un documento técnico que contenga:
 - * Una breve descripción de los subproblemas seleccionados y su modelado.
 - * La justificación de la estrategia algorítmica aplicada en cada caso.
 - * Un análisis de la complejidad temporal y espacial de cada solución.
 - * Una reflexión sobre los subproblemas no implementados, incluyendo un breve esbozo de cómo podrían resolverse.

- **Presentación Grupal:** Una exposición que cubra el diseño general del sistema y los resultados obtenidos.
- **Defensa Individual:** Cada miembro del equipo deberá explicar y justificar una parte del trabajo (diseño, código, análisis de complejidad).

Fecha de defensa: 18/02/26

Criterios de Evaluación

- **Funcionamiento correcto:** La solución debe pasar los casos de prueba.
- **Pertinencia de las estrategias:** La elección del algoritmo debe estar bien justificada.
- **Profundidad del análisis:** El análisis de complejidad debe ser claro y correcto.
- **Calidad de la documentación:** El informe y los comentarios en el código deben ser claros y concisos.
- **Coherencia y dominio:** La presentación y la defensa individual deben demostrar una comprensión sólida del trabajo realizado.