

## Simulador de Propagación de Información en Redes Sociales

Este proyecto tiene como objetivo simular la forma en que una información (como un rumor) se propaga dentro de una red social utilizando un grafo no dirigido. Cada nodo representa una persona, y cada conexión directa representa una relación de amistad o seguimiento. El propósito es observar y analizar el recorrido de la información y los tiempos de llegada a cada usuario.

El código está organizado en diferentes funciones, una para la generación de red, otra para el cálculo del algoritmo de Floyd-Warshall, otra para la visualización de la propagación y otra para la revisión de los parámetros de entrada. Esta estructura facilita el entendimiento y mantenimiento del código.

La primera parte genera la red social aleatoria con un número definido de nodos y conexiones. A continuación, se implementa el algoritmo de Floyd-Warshall, que permite calcular los caminos más cortos entre todos los pares de nodos. Finalmente, se realiza una simulación de propagación desde un nodo fuente y se visualiza cómo va alcanzando al resto de nodos paso a paso.

Para abordar el desarrollo del código, primero se genera un grafo aleatorio con `networkx`, luego se representa como una matriz de distancias usando `numpy`, y se aplica el algoritmo de manera iterativa. Las visualizaciones se realizan con `matplotlib`, coloreando los nodos para reflejar el progreso de la propagación.

Entre los retos previstos se encuentran asegurar la correcta actualización de los estados de los nodos en cada paso de la simulación y mantener la visualización clara y comprensible. Otro reto es el rendimiento al escalar la red a muchos nodos, dado que el algoritmo Floyd-Warshall tiene una complejidad cúbica.

El algoritmo de Floyd-Warshall utilizado en este proyecto tiene una complejidad temporal de  $O(n^3)$ , donde  $n$  es el número de nodos del grafo. Esto se debe a que, para cada nodo intermedio  $k$ , se revisa cada par de nodos  $(i, j)$ , realizando un total de  $n \times n \times n$  iteraciones.

En cuanto a la complejidad espacial, el algoritmo requiere una matriz de distancias de tamaño  $n \times n$ , lo que implica una complejidad espacial de  $O(n^2)$ . Este espacio es necesario para almacenar las distancias mínimas entre todos los pares de nodos.

Estas características lo hacen eficiente y sencillo de implementar en redes sociales pequeñas o medianas. Sin embargo, para redes de gran escala, podrían considerarse alternativas como el algoritmo de Dijkstra con cola de prioridad (heap) para calcular caminos desde un solo nodo.

En este simulador, se prioriza la claridad del modelo y la visualización educativa sobre el rendimiento, por lo que Floyd-Warshall resulta apropiado.

Las bibliotecas utilizadas son `networkx`, `numpy` y `matplotlib`. Como referencia teórica, se ha consultado la documentación oficial de estas bibliotecas y materiales sobre algoritmos de grafos, en particular del libro "Introduction to Algorithms" de Cormen et al.

Este enfoque permite no solo comprender la teoría de propagación en redes, sino también visualizar su comportamiento en tiempo real mediante simulaciones.