Act 4.3 - Actividad Integral de Grafos

Daniel Esparza Arizpe - A01637076

Diego Alberto Cisneros Fajardo - A01643454

Luis Fernando Li Chen - A00836174

Sadrac Aramburo Enciso - A01643639

Diego Michell Villa Durán - A00836723

Programación de estructuras de datos y algoritmos fundamentales

Grupo 605

Tecnológico de Monterrey

Jueves 23 de noviembre 2023

Act 4.3 - Actividad Integral de Grafos

```cpp
#include <iostream>
#include <fstream>
#include <sstream>
#include <map>
#include <vector>
#include <algorithm>

using namespace std;

// Estructura para representar un grafo
class Graph {
public:
    map<int, vector<int>> adjList;

    void addEdge(int u, int v) {
        adjList[u].push_back(v);
    }

    int fanOut(int u) {
        return adjList[u].size();
    }
};

int main() {
    ifstream file("bitacora.txt");
    string line;
    Graph ipGraph;

    if (!file.is_open()) {
        cerr << "Error al abrir el archivo." << endl;
        return 1;
    }

    while (getline(file, line)) {
        stringstream ss(line);
        string month, time, ip, reason;
        int day;
```

```
        ss >> month >> day >> time >> ip >> reason;

        // Separar la IP y construir el grafo
        stringstream ipStream(ip);
        string segment;
        vector<int> ipSegments;
        while (getline(ipStream, segment, '.')) {
            ipSegments.push_back(stoi(segment));
        }

        for (size_t i = 0; i < ipSegments.size() - 1; ++i) {
            ipGraph.addEdge(ipSegments[i], ipSegments[i + 1]);
        }
    }

    file.close();

    // Calcular el fan-out de cada nodo
    map<int, int> fanOuts;
    for (auto &pair : ipGraph.adjList) {
        fanOuts[pair.first] = ipGraph.fanOut(pair.first);
    }

    //Esta parte del código identifica el nodo con el mayor fan-out.
    //Asumimos que este nodo es el segmento inicial de la dirección IP
del boot master.
    auto maxFanOutIt = max_element(fanOuts.begin(), fanOuts.end(),
                                   [](const pair<int, int>& a, const
pair<int, int>& b) {
                                       return a.second < b.second;
                                   });

    cout << "Nodo con mayor fan-out: " << maxFanOutIt->first << "
(fan-out: " << maxFanOutIt->second << ")" << endl;

    return 0;
}
```

Conclusiones

Daniel:

La teoría de grafos es fundamental en el análisis de redes, especialmente en redes inalámbricas complejas, como las de 5G y IoT. Facilita la representación matemática de la topología de la red y permite simulaciones precisas de su comportamiento. Con la teoría de grafos, se pueden manejar redes dinámicas y densas sin modelos matemáticos complicados, proporcionando insights sobre conectividad, agrupamiento y latencia. Esto resulta en un análisis y diseño más efectivo y robusto de redes inalámbricas, utilizando métricas de grafos para evaluar el rendimiento de la red (*Graph Theory Applications in the Analysis of Wireless Networks*, s.f.).

En el contexto de este problema en específico, se utiliza un grafo para modelar y analizar direcciones IP en un archivo de bitácora, la teoría de grafos ofrece una forma eficiente de comprender las relaciones y conexiones entre diferentes segmentos de IP, lo que puede ser fundamental para identificar patrones o anomalías en la red, como la identificación de un boot master.

Diego Cisneros:

La utilización de grafos en computación sirve para muchas cosas, gracias a que es una estructura de datos que cuenta con nodos o vértices y aristas que conectan a estos nodos, son muy buenos para modelar diferentes relaciones en muchos contextos.

En el caso del problema de esta naturaleza pues el fan-out son los elementos o nodos que se pueden conectar a otro, (grafo) otro uso importante de los grafos es la optimización de por ejemplo, alguna ruta o trayectoria. Podría ayudar a la búsquedas más profundas, y aunque sean un gran número de datos, los grafos son muy buenos también gracias a su escalabilidad.

Diego Villa:

Los grafos, con su capacidad para representar relaciones entre entidades mediante nodos y aristas, son herramientas poderosas que se aplican en una amplia gama de campos. Desde modelar redes sociales hasta resolver problemas de logística y sistemas, los grafos ofrecen una manera efectiva de entender y analizar conexiones complejas. Su utilidad radica en su capacidad para representar relaciones de manera visual y estructurada, lo que permite abordar problemas de manera más clara y resolverlos mediante algoritmos específicos. En resumen, los grafos son una herramienta matemática esencial que ayuda a comprender y manejar relaciones entre diferentes elementos en diversos contextos.

Li Chen:

La organización eficiente de datos es esencial para garantizar un rendimiento óptimo y una toma de decisiones informada. En esta actividad al usar los grafos y al almacenar los datos nos ayuda a tener una gestión más estructurada y contextualizada de la información.

La importancia de utilizar grafos en este contexto radica en su capacidad para modelar y analizar relaciones complejas. Las redes, especialmente en el ámbito de las direcciones IP, suelen tener una estructura jerárquica y relacional que se adapta perfectamente a la representación gráfica. Los grafos permiten visualizar conexiones, identificar patrones y optimizar la búsqueda de información, lo cual es crucial en la gestión de grandes conjuntos de datos. Además, los algoritmos de grafos pueden utilizarse para realizar análisis de conectividad, identificar posibles cuellos de botella y mejorar la eficiencia operativa de la red.

Sadrac:

Utilizar listas de adyacencia para almacenar direcciones IP en un grafo proporciona una representación estructurada y visualmente comprensible de la red. Esto no solo facilita el análisis y la comprensión de la topología de la red, sino que también brinda herramientas

poderosas para la detección de patrones, la optimización de la red y la identificación de posibles amenazas o anomalías.

Referencias

Graph Theory applications in the analysis of wireless networks. (s.f.). Frontiers. https://www.frontiersin.org/research-topics/52393/graph-theory-applications-in-the-analysis-of-wireless-networks