Report

Semester Project Part-1

[Title]

[Computer Networking]

[2024]

[68022 || 66759 || 67607]

[Shahzar Hassan || Tasmia Ali || Zahra Jaffer]

[2nd]

Submitted to:

[Miss Poma Panezai]

[Department of Computer Science]

Faculty of Information and Communication Technology (FICT), BUITEMS, Quetta

Team Members and Their Responsibilities:

1. **Shahzar Hassan: CMS-ID: 68022**

He was primarily responsible for developing the project's code. Additionally, he compiled and edited the final demonstration video, ensuring clarity and completeness.

2. **Tasmia Ali: CMS-ID: 66759**

She collaborated in preparing the written report, assisting with content refinement and formatting, ensuring the documentation met required standards

3. **Zahra Jaffar: CMS-ID: 67607**

She contributed significantly to drafting the project report, focusing on structure and content quality. She also provided the voice -over for the final video presentation.

Development Approach:

Our team adopted a structured and collaborative development approach. In the initial stage, we held discussions to divide responsibilities based on individual strengths. Shahzar took the lead in coding due to his strong programming skills. Tasmia and Zahra focused on documentation, ensuring the report was accurate, well-organized, and aligned with the project requirements. Regular team check-ins were conducted to monitor progress and address any issues. This teamwork allowed us to complete each part of the project efficiently and cohesively

Agile Practices We Followed:

We followed light agile principles throughout our project. Tasks were divided into smaller units and assigned to each member. We maintained continuous communication through group chats and quick online meetings, allowing us to stay updated on each other's progress. This iterative feedback system helped us improve quality and make timely adjustments. Through not strictly following a formal Agile framework, our flexible and collaborative work style reflected key Agile values like teamwork, adaptability; and ongoing improvement.

Abstract

This project focuses on the design, implementation, and visualization of Depth-First Search (DFS), a tree-based routing algorithm used in computer networks. The goal is to stimulate how Depth-First Search (DFS) construct a DFS traversal path (DFS traversal path) from a given graph, ensuring optimal graph traversal paths with minimal weight. The implementation is done in java, utilizing the Graph Stream library for visualization. The project includes a graphical user interface (GUI) for interactive graph manipulation, real-time algorithm visualization, and detailed logging of each step. The results demonstrate the algorithm's efficiency in finding the DFS traversal path and its applicability in graph traversal tasks scenarios.

1. Problem Statement

In computers networks, graph traversal algorithms are essential for determining the depth-based graph traversal between nodes. Efficient routing minimizes congestion, reduces latency, and enhance overall network performance. Depth-First Search (DFS). A greedy algorithm, is widely used to construct DFS traversal paths, which are foundational fore many routing protocols.

The challenges lies in understanding and visualizing the algorithm's operation, including node section, edge consideration, and DFS traversal path construction. This project addresses this challenge by implementing Depth-First Search (DFS) in java, providing an interactive platform to simulate and visualize the algorithm's step in real-time.

2. Introduction

Routing algorithms are critical in network design, ensuring data is transmitted efficiently. Depth-First Search (DFS) is particularly useful for creating DFS traversal paths, which connect all nodes in a network with the least total edge weight.

 This project aims to:

- Visualize the algorithm's operation using Graph Stream.
- Provide an interactive GUI for graph manipulation and algorithm simulation.

- Log each step of the algorithm for educational purposes.

The project serves as a practical tool for understanding graph traversal algorithms and their applications in real-world networks.

3. Literature Review

Depth-First Search (DFS), developed by Robert Prim in 1957, is a well-known solution for the DFS traversal path problem. It is widely used in network design, clustering, and approximation algorithms for NP hard problems. Key studies highlight its efficiency and applicability in various domains, including telecommunications and transportation networks.

Recent advancements in visualization tools, such as Graph Stream, have made it easier to demonstrate algorithmic concepts. This project leverages these tools to enhance understanding and engagement.

4. Methodology(Design and Simulation)

Tools and Technologies:-

a) **Programming Language:** Java

b) **Libraries:** Graphs Stream (for visualization), Java Swing (for GUI)

c) **Development Enivronment:** intellij IDEA

Implementation Steps:-

1) **Graph Representation:** The adjacency matrix represents the graph, with nodes and weighted edges.
2) **GUI Development:** A user-friendly interface allows users to input graphs manually or generate them randomly.
3) Algorithm Implementation: Depth-First Search (DFS) is implemented to recursively visit all unvisited connected nodes.
4) Visualization: Graph Stream renders the graph, highlighting the DFS traversal path construction process.

5) **Logging:** Each step (node addition, edge consideration, rejection) is logged for clarity.

Key Features:-

- Interactive graph manipulation (add\remove nodes and edges).
- Real -time visualization speed for Depth-First Search (DFS).
- Adjustable visualization speed for better understanding.

5. Result and Discussion

The project successfully demonstrate Depth-First Search (DFS):

a) Graph Visualization: The GUI displays nodes and edges, with the DFS traversal path highlighted in blue.

b) Step-by-Step Logging: The algorithm's progress is logged, showing node visit order.

c) Performance: The algorithm efficiently construct the DFS traversal path, even for larger graphs(up to 50 nodes).

Example Output:

- For a 5-node graph, the algorithm selects edges with weights 2, 3, 4,and 5, forming an DFS traversal path with a sequence of visited nodes of 14.
- The visualization clearly shows the DFS traversal path edges and the order in which nodes are related

Challenges:

- Handling disconnected graphs.
- Ensuring real-time updates during visualization.

6. Conclusion and Frame Work

Conclusion:-

The project effectively implements and visualizes Depth-First Search (DFS), providing a practical tool for understanding DFS traversal path construction in networks. The interactive GUI and detailed logging enhance the learning experience.

Future Work:-

- Extend the project to include other graph traversal algorithms (e.g, Dijkstra's).

- Add support for dynamic graph updates algorithm execution.

- Integrate network simulation features, such as packet transmission over the DFS traversal path.

7. References

- **https://www.geeksforgeeks.org/dsa/depth-first-search-or-dfs-for-a-graph/**
- **https://www.geeksforgeeks.org/dsa/spanning-tree/**
- **https://favtutor.com/blogs/depth-first-search-java**