

**PROJECT REPORT**

**Graph Theory**

**Project Name: Implementation and Analysis of Bellman Ford Algorithm**

**Section:** 5E

**Team Members:**

21K-4839 Fahad Yousuf

21K-4591 Abdul Rafay

21K-4838 Huzaifa Asad

20K-1888 Muhammad Ali

**Advisor:**

Dr. Nazish Kanwal

***1. Introduction***

The provided Python script implements a Graph Visualizer using the Tkinter library for the GUI, NetworkX for graph manipulation, and Matplotlib for graph visualization. The primary goal of the application is to visualize graphs, perform real-life problem-solving using graph algorithms, and provide a user-friendly interface for graph operations.

***2. Overview***

***2.1 Graph Representation***

* The script uses the NetworkX library to represent and manipulate the graph.
* The graph can be either undirected or directed based on user selection.

***2.2 GUI***

* Tkinter is employed to create a graphical user interface.
* The GUI includes options to generate predefined graphs, run the Bellman-Ford algorithm, solve a real-life problem, and perform various graph operations (add/remove nodes and edges, save/load graphs, clear the graph, etc.).

***2.3 Bellman-Ford Algorithm***

* The Bellman-Ford algorithm is implemented for finding the shortest paths in a weighted graph.
* The algorithm handles negative weight cycles and provides a visualization of the algorithm's execution.

***2.4 Real-Life Problem Solving***

* A sample real-life problem is provided as an example, where the optimal path in a graph is found.

***3. Code Structure***

***3.1 Class Structure***

* The main class is **GraphVisualizer**, responsible for initializing the GUI and managing graph-related operations.

***3.2 Methods***

* Methods include graph generation, Bellman-Ford algorithm execution, real-life problem solving, and graph manipulation functions.

***3.3 GUI Components***

* The GUI includes buttons for various operations, a canvas for graph visualization, and a matrix frame for displaying the adjacency matrix.

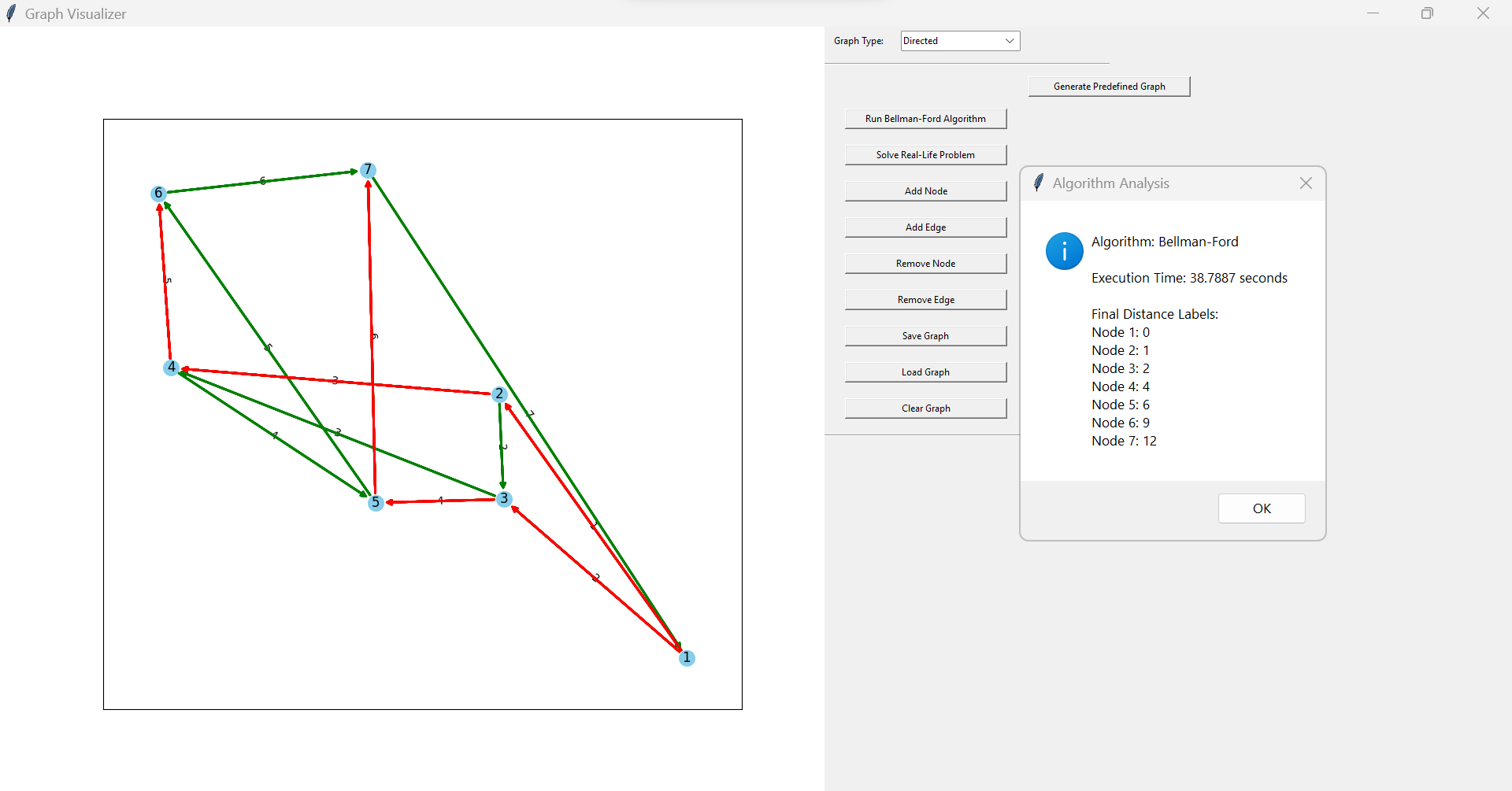
***4. Graph Visualization***

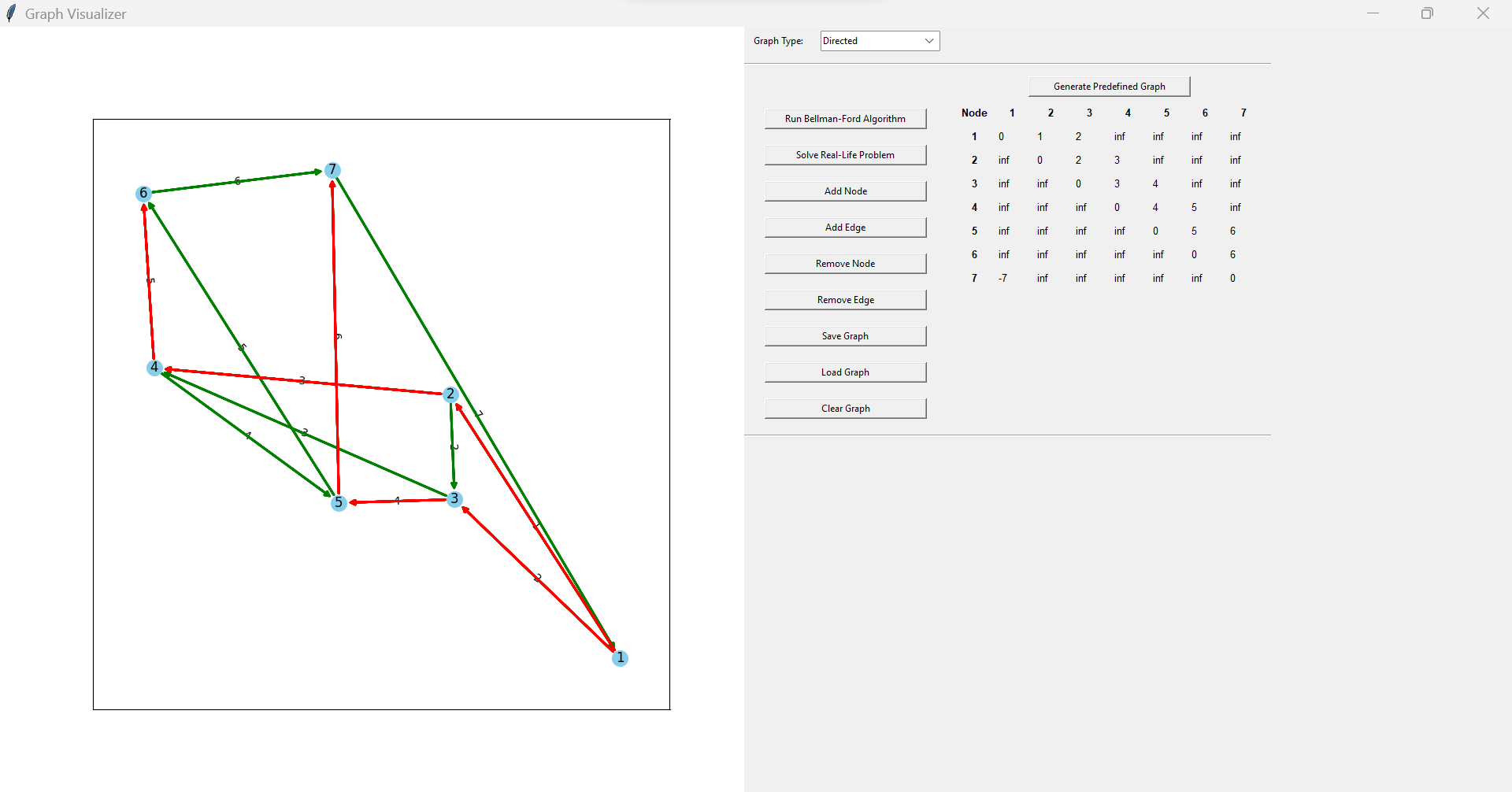
***4.1 Matplotlib Integration***

* The Matplotlib library is integrated into the Tkinter GUI using **FigureCanvasTkAgg** for graph visualization.

***4.2 Dynamic Visualization***

* The script dynamically visualizes the execution of the Bellman-Ford algorithm by highlighting edges during each iteration.

****

******

***5. User Interaction***

***5.1 Dialogs***

* The application uses dialogs (message boxes and input dialogs) to interact with the user for input and information display.

***5.2 Real-Life Problem Solving***

* An example of solving a real-life problem using the graph is provided, demonstrating the practical use of the application.

***6. Conclusion***

The script successfully combines Tkinter, NetworkX, and Matplotlib to create an interactive tool for graph visualization and manipulation. It offers a user-friendly interface for both graph enthusiasts and those interested in practical applications of graph algorithms. The inclusion of the Bellman-Ford algorithm and real-life problem-solving adds educational and practical value to the tool.

***7. Future Enhancements***

* Incorporate more graph algorithms for a comprehensive tool.
* Improve user feedback during algorithm execution.
* Enhance error handling for invalid user inputs.

Overall, the code serves as a foundation for further development and exploration in the field of graph theory and algorithms.