**Project Proposal: Eulerian and Hamiltonian Path Detection**

**Group Members: 21K-3330, 21K-3200, 21K-3201.**

**Project Overview**

In this project, we propose to create an interactive Python program using the Pygame library to detect and visualize Eulerian and Hamiltonian paths in graphs. Eulerian paths visit each edge exactly once, while Hamiltonian paths visit each vertex exactly once. The program will allow users to input a graph, and it will determine if a Eulerian or Hamiltonian path exists and display the path if found.

**Project Features**

Graph Input: Users can input a graph by specifying the vertices and edges interactively within the Pygame window. They can click to add vertices and drag between two vertices to create edges. Edges can also be assigned weights.

* Path Detection: The program will use algorithms to detect Eulerian and Hamiltonian paths. For Eulerian paths, the algorithm will check whether the graph is connected and if the degrees of all vertices are even (or exactly two vertices with odd degrees). For Hamiltonian paths, a backtracking algorithm will be implemented to search for the path.
* Interactive Visualization: The Pygame window will display the graph and the detected path if it exists. Users can interact with the graph, move vertices, and observe how the path is being determined in real-time.
* User Guidance: The program will provide information and guidance to the user, indicating whether a Eulerian or Hamiltonian path exists and, if found, what the path is.
* Graph Editing: Users can add vertices and edges, and re-run the path detection algorithm to see how the changes affect the results.

**Technologies**

* Python: The core language for implementing the algorithms and user interface.
* Pygame: The Pygame library will be used for graphical visualization and user interaction.
* Graph Algorithms: Implement Eulerian and Hamiltonian path detection algorithms.
* Graph Data Structures: Implement data structures to represent the graph, vertices, and edges.
* User Interface Design: Create an intuitive and interactive graphical user interface using Pygame.

**Conclusion**

This project aims to create an engaging and educational tool for graph theory enthusiasts and students. It will help users understand and visualize Eulerian and Hamiltonian paths while learning how these paths can be detected. The combination of Python and Pygame provides an interactive and visually appealing platform for exploring these concepts.

By the end of this project, users will have a functional tool that allows them to experiment with graphs and gain a deeper understanding of Eulerian and Hamiltonian paths.