****

**Graph Theory Project Proposal**

***Title: Implementing and Optimizing Augmenting Path Algorithm for Maximum Matching in Bipartite Graphs***

**Team Membrs:**

* Rana Wahaj Ahmed (21k-3281)
* Muhammad Hassan (21k-4577)
* Ibad ur Rehman (21k-4652)

**1. Introduction:** This project aims to implement and optimize the Augmenting Path Algorithm in both C++ and Python to find maximum matching in bipartite graphs. The algorithm's applications include resource allocation, network flow optimization, and task assignment scenarios. The transition to Python allows for a more intuitive and visual representation of the graph, overcoming limitations of console-based visualization. The project will explore the algorithm's efficiency, scalability, and its significance in graph theory and combinatorial optimization.

**2. Problem Statement:** The project addresses the challenge of finding maximum matching in bipartite graphs using the Augmenting Path Algorithm. It aims to optimize the algorithm's efficiency, leveraging Depth-First Search (DFS) to identify augmenting paths and enhance resource allocation, task assignment, and network flow optimization.

**3. Objectives:**

* Implement Augmenting Path Algorithm in C++ and Python.
* Visualize bipartite graphs and maximum matching edges in Python for better understanding.
* Optimize the algorithm using DFS to improve path-finding efficiency.
* Explore real-world applications in resource allocation, task assignment, and network flow optimization.
* Analyze the space and time complexity of the algorithm in different scenarios.

**4. Methodology:**

* Implement Augmenting Path Algorithm in C++ and Python to find maximum matching.
* Visualize bipartite graphs using Python's graphical capabilities.
* Optimize the algorithm using DFS to improve path-finding efficiency.
* Analyze and compare the performance of the algorithm in different scenarios, such as small-scale and large-scale graphs.
* Showcase real-world applications of the algorithm through case studies and practical examples.

**5. Expected Deliverables:**

* C++ and Python implementations of the Augmenting Path Algorithm.
* Visual representation of bipartite graphs and maximum matching edges using Python.
* Optimized algorithm demonstrating improved efficiency in path-finding.
* Detailed analysis of space and time complexity in various scenarios.
* Case studies showcasing real-world applications and practical significance.

**6. Conclusion:** This project aims to demonstrate the practical significance of the Augmenting Path Algorithm in solving optimization problems in various domains. By implementing, optimizing, and visualizing the algorithm, we seek to provide a comprehensive solution to maximum matching in bipartite graphs, contributing insights to graph theory, combinatorial optimization, and real-world applications in resource allocation and task assignment.

**8. Expected Visual Output:**

