Project Title: Codes for Applications of Graph Theory

Group Member: Group of three students (Preferably with the same section of students).

Programming Language: Python / MATLAB/ Sage Math.

Project Weightage: 08 Marks

Instructor: Dr. Nazish Kanwal Sections: BCS-5A, 5C & 5E.

Objective:

The goal of this project is to evaluate and improvise the codes for different graphing algorithms and applications of graphs in Computer Science, deep learning, and natural sciences. Additionally, you may check the efficiency of code in two different programming languages for a selected problem. Students will analyze and improve upon previously defined codes, providing insights into the advantages and disadvantages of codes.

Instructions:

1. Problem Statement:

You must select a problem related to graph theory that requires algorithmic processing. You must solve this problem and implement code in programming languages of your choice.

2. Code Implementation:

You can start by referring to previously defined codes, if necessary, but ensure to write your code from scratch.

- Implement the solution using one programming language.
- Document the code clearly with comments explaining the logic and any optimizations you make. Ensure that your code is efficient, and it produces the correct output.

3. Second Language Implementation (In case of efficiency check as well):

- Repeat the process by implementing the same solution using a different programming language.
- Document this code as well, emphasizing any language-specific features or challenges.
- Ensure that both implementations provide the same correct output for the given problem.

4. Efficiency Analysis:

- Run both implementations with various input sizes to analyze their performance.
- Measure and compare the execution time, memory usage, and any other relevant metrics.
- Provide insights into each language's strengths and weaknesses for solving this problem.

5. Code Output:

- Generate the output of your code for specific test cases (number of vertices not less than 10).
- Print the output as a hard copy, ensuring that it's legible and well-organized.

6. Submission:

- Submit your code electronically in soft copy format on GCR.
- Submit the hard copy of your code's output along with any analysis or insights you've gained.

7. Presentation:

- Prepare a brief presentation to share your findings and comparisons with the class.
- Be ready to answer questions and explain your approach.

Grading Criteria:

- Correctness of code implementation.
- Efficiency and performance analysis.
- Documentation and comments.
- Output presentation and analysis. (Drawing of the graph with labeled vertices and highlighting your output via specific color.)
- Presentation clarity and engagement.

Important Dates:

- Proposal Submission Date: 24th October 2023.
- Project Submission Deadline: 1st December 2023.
- Presentation Date: 30th November 2023.

Notes:

- Plagiarism will not be tolerated. Ensure that your code is original.
- Seek help from your instructor or classmates if you encounter difficulties.