

Project 3 Team: Arshia, Mahan,Saba, Pariya, AmirMahdi

Classes and Responsibilities

1. Graph Class Implementation

- Responsibilities: Core functionality for graph representation, loading from CSV/JSON, converting to different formats, and executing algorithms like BFS, DFS, and Dijkstra's algorithm.

2. File Handling and Utility Functions

- Responsibilities: Helper functions for file operations, including reading and writing CSV/JSON files.

3. Testing and Validation

- Responsibilities: Create test scripts to validate the functionality of the Graph class, including loading and saving data, and running graph algorithms.

4. Documentation and Integration

- Responsibilities: Document the Graph class and its methods, write usage instructions, and ensure that all components work together.

5. Advanced Features and Optimization

- Responsibilities: Implement advanced features, optimize algorithms, and handle performance improvements.

Assignments

1. Arshia: Graph Class Implementation

- **Responsibilities:**

- Implement the Graph class with methods for:

- Initialization and basic setup.
- Loading graph data from CSV and JSON files.
- Exporting graph data to CSV and JSON.
- Implementing BFS, DFS, and Dijkstra's algorithm.

2. Mahan: File Handling and Utility Functions

Responsibilities:

Develop helper functions for:

Reading and writing CSV files.

Reading and writing JSON files.

Validating file formats and handling errors.

3. Saba: Testing and Validation

• Responsibilities:

- Write and execute test cases for:
 - Loading from CSV and JSON.
 - Exporting to CSV and JSON.
 - BFS, DFS, and Dijkstra's algorithm.
- Ensure correctness and handle edge cases.

•

4. Pariya: Documentation and Integration

Responsibilities:

Document the Graph class and its methods.

Write usage instructions and prepare a README file.

Ensure integration of all components and prepare the project for submission.

5. AmirMahdi: Advanced Features and Optimization

- **Responsibilities:**

- Implement advanced features such as:
 - Enhancing performance of BFS, DFS, and Dijkstra's algorithm.
 - Adding new functionalities like shortest path between two nodes, detecting cycles, etc.
- Optimize existing methods for efficiency and handle large graphs.