Floyd's Algorithm Implementation

This repo shows the work of Floyd's algorithm to find the shortest paths between all pairs of vertices in a given graph. With both iterative and recursive implementations of the algorithm.

* Installation

To use this project, you must install Python on your system. Clone the repository to your local machine:

```bash

git clone https://github.com/your\_username/floyds-algorithm.git

Navigate to the project directory:

cd mid\_module\_assignment

* Usage

You can use both the iterative and recursive implementations of Floyd's algorithm by importing the corresponding functions from the mid\_module\_assignment module.

* Create a graph (adjacency matrix)

graph = [

[0, 7, NO\_PATH, 8],

[NO\_PATH, 0, 5, NO\_PATH],

[NO\_PATH, NO\_PATH, 0, 2],

[NO\_PATH, NO\_PATH, NO\_PATH, 0]

]

# Use the iterative implementation

floyd\_algorithm(distance)

# Use the recursive implementation

floyd\_algorithm(graph)

* Running Tests

To run unit tests, execute the following command:

pytest

To run performance tests, execute:

pytest --benchmark-autosave

## License

This project is licensed under the MIT License - see the LICENSE file for details.

In this README file:

- Installation instructions guide users on how to clone the repository, install dependencies, and set up the project.

- The usage section demonstrates how to use the implemented functions in Python code.

- Running Tests section explains how to run unit tests and performance tests using pytest.

- License section provides information about the project's license.

floyd\_algorithm/

├── iterative/

│ ├── iterative.py

│ ├── tests/

│ │ └── iterative + unittest with caverage.py

│ ├── performance/

│ │ └── performance\_test\_iterative.py

│

├── recursive/

│ ├── recursive.py

│ ├── tests/

│ │ └── recursive +unittest with coverage.py

│ ├── performance/

│ │ └── performance-test-recursive.py

│

├── requirements.txt

└── README.md