**Overview**

In the world of ridesharing, where efficiency and speed are paramount, the application of advanced algorithms can revolutionize operations. This project, a cornerstone of the Advanced Algorithm and Complexity course, delves into various algorithmic strategies, applying them to a real-world ridesharing scenario. The essence of the project lies in its practical approach, using a rich dataset to demonstrate the efficacy of these algorithms in optimizing ridesharing logistics.

**Data Quality and Preprocessing**

The foundation of any data-driven solution is the quality of the data itself. The initial phase of this project focuses on assessing and refining the dataset to ensure its reliability. Through careful preprocessing, we ensure that the data is clean, relevant, and structured, setting the stage for accurate and insightful algorithmic applications.

**MergeSort and QuickSort**

At the core of data manipulation are sorting algorithms, which are pivotal in organizing and analyzing large datasets. In this project, two classic sorting algorithms, MergeSort and QuickSort, were implemented and compared. By applying these algorithms to the 'Demand' column of the dataset, we observed that QuickSort outperformed MergeSort in terms of execution time. This finding is significant in a ridesharing context, where rapid data processing can lead to more efficient decision-making.

**Divide and Conquer**

The divide and conquer strategy, a fundamental approach in algorithm design, was explored to break down complex problems into simpler, more manageable sub-problems. This methodology is particularly effective in handling large-scale datasets common in ridesharing scenarios, enabling more efficient data processing and analysis.

**The Nearest Neighbour Heuristic Greedy Algorithm**

One of the highlights of the project is the application of the Nearest Neighbour Heuristic Greedy Algorithm. This algorithm, often used in solving the Traveling Salesman Problem (TSP), was adapted to optimize drivers' routes in the ridesharing model. By sequentially selecting the nearest unvisited customer, the algorithm minimizes travel distance or time, thereby enhancing operational efficiency, reducing fuel consumption, and improving customer satisfaction.

**Greedy Algorithm for Optimized Ride Assignment**

Efficiently matching drivers with customers is crucial in ridesharing. This project implements a Greedy algorithm that assigns the closest available driver to each customer. By prioritizing proximity, the algorithm effectively reduces waiting times and travel distances, leading to an optimized assignment of rides.

**Travel Salesman Problem (TSP)**

The TSP, a well-known problem in computer science, is adeptly applied to the ridesharing domain. The project demonstrates a method for calculating a distance matrix between locations and employing a nearest neighbor heuristic to approximate a solution to the TSP. This approach is vital in devising efficient routes for drivers, ensuring that each passenger is picked up and dropped off in an optimal sequence.

This project not only demonstrates a thorough understanding of various advanced algorithms but also highlights their practical application in a real-world context. The algorithms explored and implemented offer significant improvements in the operational efficiency of ridesharing services, a testament to the power of algorithmic solutions in addressing contemporary challenges. Through this endeavor, we gain invaluable insights into the dynamic and ever-evolving field of algorithm design and its profound impact on modern-day industries.

## Contributing

We welcome contributions to the Air Quality Monitoring at CCI project! By contributing, you can help improve the project and make it more useful for everyone. Please follow these guidelines to ensure a smooth contribution process.

**How to Contribute**

1. **Fork the Repository:**

* Click the "Fork" button at the top right corner of this repository page to create a copy of this repository under your own GitHub account.

1. **Clone the Forked Repository:**

* Clone your forked repository to your local machine using the following command:

bash

Copy code

git clone https://github.com/yourusername/Air-Quality-Monitoring-CCI.git

1. **Create a Branch:**

* Create a new branch for your feature or bug fix:

css

Copy code

git checkout -b feature-or-bugfix-name

1. **Make Changes:**

* Make your changes in the new branch. Ensure that your code is well-documented and follows the project's coding standards.

1. **Commit Changes:**

* Commit your changes with a clear and concise commit message:

sql

Copy code

git commit -m "Description of the changes made"

1. **Push Changes:**

* Push your changes to your forked repository:

perl

Copy code

git push origin feature-or-bugfix-name

1. **Create a Pull Request:**

* Go to the original repository on GitHub and create a pull request. Provide a detailed description of the changes you have made and why they should be merged.

**Code of Conduct**

Please note that this project adheres to a Code of Conduct. By participating, you are expected to uphold this code. Please report unacceptable behavior to [e.nwonye0320231@arts.ac.uk].

**Reporting Issues**

If you encounter any issues or have suggestions for improvements, please feel free to open an issue in the GitHub repository. Provide as much detail as possible to help us understand and address the issue promptly.

**Feature Requests**

We welcome new feature requests! If you have an idea for a feature, please open an issue and describe your proposed feature in detail. We will discuss the feasibility and implementation plan with you.

**Testing**

Before submitting your changes, please ensure that you have tested them thoroughly. Include unit tests or integration tests as appropriate to ensure that your changes do not introduce any new issues.

**Documentation**

If your changes involve significant modifications or additions, please update the project documentation accordingly. This helps others understand the new features and how to use them.

Thank you for your contributions! Together, we can make this project better and more impactful.

## License

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

## Contact

For any inquiries or feedback, please contact:

* Emmanuel Nwonye
* Email: e.nwonye0320231@arts.ac.uk
* LinkedIn: <linkedin.com/in/emmanuel-nwonye-40a024183>