First of all, we'd like to thank you for choosing Loadsmart. We hope this email finds you well. As a next step in the recruitment process, we have an exercise for you.

This is an opportunity to show your back end skills. Just be clear and do it to the best of your ability - the faster you complete it, of course, the quicker we can move forward.

Goal

Given a list of trucks and their current locations and a list of cargos and their pickup and delivery locations, **find the optimal mapping of trucks to cargos to minimize the overall distances the trucks must travel**.

Please assume:

1. Each truck can only carry up to one cargo.
2. Each truck can only make up to one trip.
3. Some trucks may not be used at all.

Please find attached 2 .csv files for you to complete the assignment:

- cargo.csv: a file containing a list of cargos we need to move (with product name, origin and destination city).
- trucks.csv: a file containing a list of trucks and their home city.

Please specify which Python version you have used to run your solution.

Solution

Your solution should be achieved by running the following command:

```
$ cargo_truck <cargo_file> <truck_file>
```

For example:

```
$ cargo_truck data/cargo.csv data/truck.csv
```

The solution should generate a `results.csv` file in the root of the directory.

The first row is the header `cargo, truck, distance`.

The following rows are the solution with the following format:

```
cargo, truck, distance
```

Where:

- cargo: the cargo that will be moved.
- truck: the truck that will move the cargo.
- distance: how many **miles** the truck will travel, from truck's origin to cargo's origin and to final destination. The distance must consider only one trip, not a round trip. The number must be a float round up to two decimal places.
- Rows must be sorted by distance (ASC)

Example:

```
cargo, truck, distance
Oranges, Sidhu Trucking Incarrisburg, 1000
Apples, Jorge L Denisollywood, 2000.05
```

Documentation

Your solution must have documentation, that should contain:

- Instructions on how to install the solution
- Instructions on how to run the solution
- Instructions on how to run the tests

Also, you might want to add:

- Code documentation
- Decisions you made during the development
- Any other information you feel might be useful for the reviewers

What we are going to assess

- Solution correctness.
- Project structure and architectural aspects.
- Unit tests are mandatory. Please, follow the best practices you already know to build them.
- Code quality, simplicity, and readability.
- Documentation: the quality of it, if it is clear, objective, and useful.
- Algorithm complexity. How the algorithm runs.

This will count as a major part of your recruiting process, so please know the gravity of this assignment.

Your code should be tested. You can use unit and/or integration tests. Just build the tests you think are important for the proposed challenge.

Please do not use a library ready function that solves the problem. We want to evaluate your algorithm building skills.

Although you can use another language, we strongly suggest that you use Python for this exercise.

When finished:

- Email the **ZIP** solution to task@loadsmart.com
    - Please label the email subject line (Your Full Name - Loadsmart Back End Test).

- Keep in mind that we may ask you to improve and/or change your solution during the interview.

- If the file is too big, try sharing it through a private repository (like Bitbucket or GitHub) or just share it using Google Drive.

- Please, **do not post your solution to any kind of public repository**.