First of all, we'd like to thank you for your interest in Loadsmart. We hope this email finds you well. As the next step in the recruitment process we wanted to do a quick exercise with 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 that each truck can only carry up to one cargo, each truck can only make up to one trip and that some trucks may not be used at all.

Here are 2 csv files for you to complete the assignment: cargo.csv and trucks.csv. Cargo.csv is a list of cargos we need to move (with product name, origin and destination city) and trucks.csv is a list of trucks and their home city.

Please insert a comment (or docstring) at the beginning of the file (before line 5) saying which Python version you have used (e.g.: 2.7, 3.6, etc)

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 repo (like bitbucket) or just share it using googledrive.
- Please, **do not post your solution to any kind of public repository**.

Important things:

- This will count as a major part of your recruiting process so please know the gravity of this assignment.
- Loadsmart will reach out to you once the test is reviewed.
- In this test, we are interested to understand how you create good algorithms more than what technologies you used. So just printing the output is ok (you don't need a database or a UI, but you can create one if you want).
- For the same reason explained above, **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.
- Your code should be tested. You can use unit and/or integration tests. Just build the tests you think that are important for the proposed challenge.

What we are going to assess:

- Solution correctness
- Project structure and architectural aspects
- Automated tests (they are mandatory to be written following the best practices you already know)
- Code quality, simplicity and readability
- Documentation (we don't want verbose texts, but other developer needs to read the minimum instructions to run your solution successfully)
- Algorithm complexity: how it runs