

# ALGORITHM ENGINEER.

## Assignment.

### Purpose

The purpose of this assignment is to gauge:

- Your reasoning with regards to the choice of algorithm and your knowledge of the field.
- Your creativity in the approach.
- Your software development skills.
- Your skill with modeling.
- Your ability to report and explain your work (this is mostly in the interview afterwards).

### The problem

For this problem, an event is a location on a 2-dimensional plane, encoded by its coordinates ( $x, y$ ). You are given two sets of events, a set of deliveries and a set of pickups. Let's call them D and P, respectively.

Your task is to create a tool that:

- Is given the sets D and P as input (encoded and input however you want), and outputs a “reasonably short route”.
- Each event has a required capacity. The vehicle handling the events has a physical capacity limit. The capacity of events in the vehicle at any given time may not exceed this physical limit.
- A route is an ordering of events that starts and ends with a “depot” event, located at the origin (0, 0). The depot should not be visited in the middle of a route.

- The length of a route is calculated as the sum of distances between subsequent events.
- A delivery is loaded onto the vehicle at the first depot and dropped off at its specified location. A pickup, conversely, is loaded at its specified location and unloaded upon returning to the depot at the end of the route.
- The twist: the route should visit ***as many delivery events as possible*** but only one (1) pickup event *of your choice!* The route length is therefore a secondary objective.

You can think of this as the famous traveling salesman problem except for the twist.

## Remarks

- We are aware that the problem is *hard*, and that depending on the size of the input, it is infeasible to calculate the optimal route in a reasonable amount of time, for some of the objectives. You are free to use approaches that only approximate the optimal solution (but you have to be prepared to explain why it is good, both in the interview, and as some written analysis included in the returned assignment.).
- We are aware that as an algorithm engineer your main expertise is in the mathematical background related to the algorithms you use, but for this position you are responsible for implementation as well, and your code goes to production. Hence, good software development skills, code hygiene, and result validation (e.g. tests) are required.
- Also, it is clear that there are many out-of-the-box packages/libraries to solve this kind of problem out there. But we want to see how *you* solve this problem. Show us your prowess!
- If you are looking for input data (feel free to generate your own!), you can find it here: <https://www.sintef.no/projectweb/top/vrptw/homberger-benchmark/>.

Keep in mind that these are full problem instances that are *not* the one asked here. You should only use the events themselves, and can change them as you see fit.

## Final note

Your work should be implemented in either Python, Rust or C# and handed-in as a github repository.

And, again, your code should be “production-ready” (hint: think of code clarity and tests). But take into account that you should spend around 4-8 hours on the assignment, so you will need to make wise choices on which corners to cut.