Casus 1 – Parcel service

A parcel service has a large fleet of vans and various drivers.
Every day, large quantities of parcels of various sizes are delivered from the distribution center to all kinds of addressees.
The status of a package to be transported is known from the moment of receipt at the distribution center to the moment of delivery to the addressee (distribution center, en route, delivered, lost, damaged, etc.).

For an efficient service, there are several problems that need to be solved, such as:

- How can you request the current status of a given package as quickly as possible?
- Who are the top 10 recipients in a recent period (day, month, ...)?
- Has all data been saved in the right data format?
- Which packages need to be transported to a specific region?
- Given a road map, what is the fastest/shortest route from the current location towards a specific client's address?
- How do you create an efficient route for a driver to multiple given addresses?
- How do you load as many packages of different sizes as possible into a van (if you are not allowed to stack), given the dimensions of the loading floor?
- If a large number of parcels of different sizes have to be transported from the collection point to one or more addresses, how do you distribute this load among one or more vans, so that as few trips as possible are sufficient?
- How many vans/drivers do you need per day

Perhaps you can think of other interesting issues that are important for the parcel service.

To solve the issues of this parcel service you need a lot of data, which you of course want to store as smartly as possible. Then different algorithms are needed to find solutions to the different problems.

For simplicity, we assume a distribution area in which we can indicate all addresses with an x-coordinate and a y-coordinate.
The road map does not have a direct road between each two addresses, there are only a limited number of roads. Furthermore, roads intersect at various intersections (also indicated with an x- and y-coordinate).
You can define different road maps yourself, where each road has a length and/or a time duration to travel it.
Below you will find an example with a distribution center (D) and various delivery addresses for inspiration.

We have the following data available for you:

- The address of the distribution center (375,375)
- An overview of clients with their address (csv-file)
- An overview of parcels with dimensions, date of entry at distribution center and corresponding addressee (cvs-file)

Develop a *proof of concept* for the parcel service for an application that solves a number of their issues/use cases. Show with the help of the given data, possibly supplemented by yourself with generated data, that the *proof of concept* works properly

In your *proof of concept* each of the following data structures and algorithms must be used:

- A linear data structure and a binary search tree, with the same data stored in both structures. In both data structures must be searched for a specific element and the required search time must be measured and compared against different (large) numbers of stored data.
- A sorting algorithm.
- A data structure that makes the desired functionality as efficient as possible.
- A graph structure with data, on which one or more graph algorithms are applied.
- Recursion and a backtracking algorithm.
- Finite automata and/or regular expressions.