# Report

# Code structure and Design choices

Garbage collection system:

- Written in Python: The reason behind this choice is mainly, that I find the language fun, and I wanted to explore some of its features.

- Databases are not used as I felt in almost all cases, data amounts are not big enough to justify creating a separate database.

Route planning:

- Dijkstra algorithm to calculate distances betweenb bins: . The reasons I chose Dijkstra’s are: First, it is relatively fast, compared to other distance calculation algorithms, its time complexity is only … . And second, it can be very conveniently used to calculate distances between any two points in a graph. I use Dijkstra to create a hashmap containing the distance between vevery tow points if the road layiut. While this may seem wasteful at first glance, for most input types it is much faster than caculationg only the requitred distances in real time.

- Nearest neighbour for route creation: algorithmsThe very simple nearest neighbour algorithm is used after that, to direct the lorry to the closes bin. While this does not always produce the most optimal route, it is, in most cases not far from it, and its shortcomings are compensated by the algorithm’s incredible simplicity and speed.

Events generation:

- The events generated during the simulation are stored and executed via a priority queue – which is very convenient, as one can use the timestamp of an event as its priority and easily execute the events with smallest timestamp first. The system uses hours as main time unit, and they are only converted in other formats when required for output.

## Tests: