ENGSCI233_PROJECT: Design_Brief_skim618/573360750

The algorithm first partitions the given list of rest-homes into 4 lists for each of the four couriers. I first sorted all the longitudes in order and split it in half and assigned it as west and east. I then sorted the two according to their latitudes and noticed that the homes in north-west are the furthest away relative to the airport, so only a third of the homes were given to the courier that took this path. The couriers who took the east side took an even amount each. These lists are then fed through the nearest neighbour algorithm to generate paths for each subnetwork. By starting at Auckland Airport, I searched for the closest home, and set this home as my new start point. Iterated this through until all the homes have been reached. The algorithm also ensures that the starting and finishing point is at Auckland Airport.

I chose this algorithm as it is easy to follow and understand, while also being *heuristic*; it only gives an approximation of the real solution. If I was to implement an algorithm that achieves 100% optimality, the total computation time would take longer and inefficient. By choosing this heuristic approach, I can save computation time and get a close approximation to the optimal solution.

The average total computation time for my desktop takes around 650 seconds (with debugging) and 235 seconds (without debugging). Distance travelled by each of the couriers are 17.0, 17.2, 13.7, 14.3 hours respectively.