

# The Traveling Zip Problem

**Expected time:** 3-5 hours

At a Zipline operations center, we want to deliver many different types of products to many different hospitals using our fleet of autonomous planes (we call them "Zips"). In order to provide the best overall experience to our customers, we always want to minimize the time-to-deliver. That said, some of our orders are "Emergency" deliveries, which have the potential to be life-saving, whereas other orders are "Resupply" deliveries, leaving us some flexibility in delivery time.

The reality is that Zips are a finite resource, and so we need to be more clever than immediately sending out a Zip as each order comes in. To help manage these logistics, the purpose of this exercise will be to write a ZipScheduler utility (preferably in python), to determine when and to which hospitals we should send out Zips for delivery.

To make things more interesting, we will also allow each Zip to carry multiple packages at a time. This means that a Zip can fly a route, delivering to several hospitals along the way, although Zips are still limited by the cumulative distance they can fly on a given trip.

## Inputs

### **hospitals.csv**

The locations of the hospitals we are delivering to are provided in a CSV file specifying:

*Hospital Name, North, East*

North and East are provided in meters with our operations center assumed to be at (0,0)

### **orders.csv**

In general operations, we will receive a sequence of orders in real-time. Each order specifies the hospital the delivery needs to go to, and the priority of the delivery. To test your system we are providing a list of orders in a CSV, specified as:

*Received Time, Hospital Name, Priority*

Received Time will be specified in seconds-since-midnight, and priority will be either 'Emergency' or 'Resupply'

## Outputs

Your primary deliverable is the implementation of a `ZipScheduler` class, which can be used as part of our delivery management system.

Please implement a class that exposes the following public APIs:

- `queue_order(received_time, hospital, priority)` is used to add a new order to our queue, and will be called every time a new order arrives.
- `schedule_next_flight(current_time)` will be called periodically (approximately once a minute), and should return either `None`, or an ordered list of hospitals to serve in the next flight.

There is no single correct answer to this problem. You should prioritize having an answer and a framework for exploring the problem over immediately worrying about the most optimal answer. We're looking for you to make reasonable tradeoffs in your approach and explain your choices.

You may assume that for a flight, the Zip flies a series of straight-line paths, starting at the operations center, passing through each hospital, and then returning home again. The total length of these paths must be less than the maximum range of the Zip. Zips must always return home before the end of their flight, but once they have returned, they charge instantly and can be immediately sent out again.

You should make your code configurable where appropriate, but some reasonable values to use for evaluation are: 10 Zips at the operations center, and each Zip can carry up to 3 deliveries, has a flight-speed of 30 m/s, and a maximum cumulative range of 160km.

In addition to the class definition, please write enough code to have confidence that your solution is reasonable.