Nowadays, transportation and deliveries are receiving a big push from the AI world. A lot of technologies, devices are introduced and some are already operational:

* Drones for deliveries
* Drones used for taxis
* Self-driving cars
* Driverless busses and taxis
* Traffic prediction
* Smart traffic lights and bus stations
* Etc.

**Homework 1**

We live in a very innovative region, a metropolitan area that currently wants to research the possibilities for creation of a drone delivery network, helping its citizens. While that may sound admirable, there are lots of unknowns in front of our municipality to start project planning. Fortunately, the mayor is an urban and metropolitan studies graduate, therefore it can be easily seen that this is a tough optimization problem. It hides different challenges that require a diverse skillset to solve them accordingly:

1) how many warehousing facilities would be enough to serve potential customer orders,

2) how to strategically choose a warehousing facilities location,

3) how to find some delivery time estimate, depending on the warehousing facilities distribution in the area,

4) how to plan the optimal delivery routes in the metropolitan area with lots of horizontal and vertical construction - how to take off and land a drone in this area,

5) how to plan the delivery time needed, avoiding all obstacles - is there a way to precompute all the possible routes apriori to instruct the drones easily or a complex type of drones will be required to be able to navigate optimally in the metropolitan area.

Even though we are not experts in math, statistics and machine learning, nor do we have solid experience in urban planning, we know that we have to split the problem to a smaller ones. Having that argument, a different dimensions of the problem can be simplified, substituted, approximated first.

Let's focus over the drone movement since it is not always easy to avoid obstacles, plan the entire path and time needed. A very good place to start is to ignore all the path obstacles in the 3D world. One step further into this direction dictates that the region map can be substituted with its 2D representation. A location representation becomes a 2D point (a point having two coordinates). The starting problem is to invent a way to know the exact path a drone will go through all the obstacles (and time needed) delivering a package from a warehouse to a building entrance for a customer. For the purposes of our municipality doing an initial research and project planning, а reasonable simplification of the problem is to find a straight path without all obstacles.

To put it differently, a drone moves only in straight lines between a current position and a target (a customer or warehouse location) and then picks another target. Another thing can be concluded from that thinking about the metropolitan area - it can be substituted with a rectangular representation and all drone movements can happen in that area. A customer as well as a warehouse location becomes a point the 2D representation of the metropolitan area.

Now let's continue delving into about another aspect - the time. Even if we "ignored" the obstacles to make the problem 2D, the movement still takes time. A few examples where the drones can wait are:

* at the customer location, till the order is picked up,
* potentially waiting in the warehouse to take the next customer delivery,
* to be re-charged,
* to not have a constant movement due to normal movement conditions or time needed to recompute a route or other thing that make the drone wait instead of moving at a constant speed,
* different drones might have complex motion algorithms requiring intermediary stops due to weather conditions, battery charge or other reasons.

For helping the municipality with the initial research, a rough estimate of delivery time is enough. Therefore the movement speed can be considered constant and equal to a single unit in the map for a minute (for example, the drone travels the distance from point (0,0) to the point (0, 1) for one minute). A significant research have to be made about the types of drones, their prices and other characteristics like the battery capacity and consumption. Again, for the rough estimate of time, it is OK to assume that drone batteries are enough to do deliveries during the day and are recharged during the night, inside warehouses (that also means that our drone delivery network will deliver orders only during the day).

As for the time needed to stay at warehouse to pick the next order we can assume that this is about 5 minutes and this is the time just for picking the order, which will be prepared in a package beforehand. The order, when delivered could be just left in front of the customer's door and a separate notification is sent, meaning there is no waiting time for the drone, to wait the the customer to pick up the order.

Thinking about product storages inside the warehouses, the product inventories, the complex system needed to fill the warehouse with needed products is a separate complex problem we can split out of the original one and leave it unsolved for now. For now, a good approximation we need to consider is that all the warehouses are having unlimited quantities from all the products.

Last but not least, at any given time, a drone can carry only a single delivery of all the products inside it.

Even with all the assumptions the mayor needs our help developing a program that can help planning the establishment of a drone delivery network, the needed resources investment and potentially the stages needed in this project to start and complete it.

**a) (Basic Drones - time)**

Our first task is to deliver an application to the mayor that, given the warehouses and customer locations as well as a list of different customer orders, returns:

* the total time, in minutes, needed to deliver all orders (the minute that is considered as last is the one in which the last order is delivered and dropped at a customer location), not taking into account the time in which all the drones go back at a random warehouse to recharge during the night.

Having this application, the municipality representatives can play with it, enter different input, different places to position precisely the warehouses, different numbers of orders, products, in order to get an understanding of what can be achieved and plan carefully how to proceed with this project.

**b) (Basic Drones - drones used)**

Together with the total time, the number of drones that are used to do those deliveries is needed as well in order to estimate the resources needed to establish that delivery network.

**c) (Drone charging - consume only when move)**

The current application is useful for the municipality to start the work on the project to have drone deliveries in the cities around. However, having the initial very rough estimates led the municipality representatives start communication with companies that sell drones in order to get to know more about prices and how they are used, maintained, etc. From it, they understood details about drones that have to be taken into consideration in the application we have built. The drones that have enough capacity to last a full day are very expensive. Fortunately, there are more affordable alternatives that are offered to the municipality. The ones that deserve attention, having in mind their prices and maintenance, are having big enough batteries to last a few deliveries but they have to be recharged. Their power/battery consumption is based on the distance they cover. If it happens that their battery is drained when moving on the map, the drone is taken back to the warehouses at night. We have to develop an extension of the application (it still needs to return the things from the previous experiments) in which we can configure:

* The battery capacity of each drone, in kW,
* The drone power consumption, in W per minute.

It is safe to assume a few things on top of the previous:

* The drones are still recharged at night. During the day, those drones we consider using, can be recharged at the warehouses (we will install chargers inside them) and each full charge takes 20 minutes (proportionally for not a full charges),
* The capacities and consumption can be found in the example below
* If there are drones out of capacity on the map, they have to stay there till the end of the day. Since we rely on the quality of the products we can consider the load of such drones as lost.

You can have a look at the few example inputs [here](https://docs.google.com/document/d/1E06mmE0O-xGv4hx3BS4c8F9mIIBayX4nP0i2FMq2J30/edit?usp=sharing), finding warehouse and customer coordinates, product names, order lists, etc.

Using the program for doing rough estimations of the project, new ideas popped up that will be even make it more useful. A more realistic scenario is to be able to see the entire flow of the program, all the movements of all the drones in order those orders to be delivered (can be simple console output).

**d) (Program output real-time)**

The program, having the same goals already implemented, should be able to configured to play X program minutes for Y real-life milliseconds. That configuration should be given in the input, together with the rest of the unknowns - warehouses, locations, orders, etc. The program should also be adjusted in such a way that it can be configured to run with and without output (without is running the functionality from the previous day, directly outputting the result without intermediate drone movements and orders fulfilled).

**e) (Add new order)**

Moreover, to add more realism into the program (if it is configured to run with a real-time output), a customer should be able to enter a new order while it is running. The easiest way for a user to enter an order is through the same configuration file (it can be read every program minute for making it user friendly).

**f) (Basic Drones - average time)**

The municipality experimented a lot with the current program and an initial estimate of the resources is done. However, a next phase of the project to create a drone delivery network is to delve more into the details about precision of those deliveries, optimal time needed, types of drones used to make that time optimal, algorithms and movement advancements they have, their price, etc. To start with that, a good metric about deliveries which is helpful in project planning is to return the average time for an order to be delivered (with the total time and number of drones needed). That means, the program that currently is outputting the number of drones used to deliver the orders and the time needed have to output also the average time of delivering a single order.

A viable next phase of the project the planned activities include manual testing from different users who should be able to work with the program simultaneously in order to add orders and gather more insights about the drone delivery network. The best would be to create a new program that is working together with the previous one (sending to it what the users wants and returns an output). Differently said, if the first program is configured to run with real-time output then a second program should be able to connect to it while sending commands and receiving results. In this way, different people can start separate testing creating real-life scenarios.

**g) (Add new Order)**

There should be a way in the second program, to enter a new order (one or more) through a configuration file. For simplicity, let's reuse the same configuration format from the first program and the logic to be read every program minute in order to monitor for a new commands from the user. To restate it once again, the goal is to run the first program in real-time mode and a few users to be able to run separate instances of the second program in order to add new orders whenever they want.

**h) (All orders status)**

There should be a way in this second program to see statuses of all orders (to be delivered, already delivered, currently in delivery). To reuse the configuration file for adding commands, the best is to make a on/off switch in it such that when turned on the second program should output all order statuses. Again for simplicity, it is best to have one more configurable property for the frequency the statuses should be output. The value of that frequency property should be in program minutes, meaning the interval in which the status output should be delivered (if the switch is on). To summarize, different users will start the second program, (add orders), and configure it to start outputting status updates during some time interval.

**i) (Drone charging - consume for each kilogram taken)**

A new type of drone is offered to the municipality - its power/battery consumption is based not only on the distance covered but also on the carried load. Obviously we want to accommodate that in the program (it still needs to return the things from the previous experiments) in which we can configure:

* The weight of each product, in grams,
* The drone power consumption, in W per kilograms per minute.

It is safe to assume a few things more (to make the program easier to work with):

* The weight a drone can carry is limited only from its power consumption,
* The exact power consumption in each minute is calculated as rounding the current load to kilograms,

**j) (Charging stations network)**

Continuing the communication with companies selling drones it turns out they sell charging stations as well. That sounds like something which again needs to be explored if it helps the deliveries somehow and what would be its cost. There are multiple offers which the municipality has to consider and they include 3 types of charging stations - cheapest, normal, fast (they differ in price and time they charge) and 2 types of drones - with normal batteries and fast-charged ones. Our application has to accept the following input:

* the location of the drone stations we want to use with their type
* The type of each drone

The assumptions - a single drone could be charged at any time in a single station (the others will potentially have to wait).