Q2

The company decides to collaborate with an airline to facilitate deliveries. This airline owns multiple airports located in this region. A truck can take a flight from one airport to any other airport. For each flight, a plane may only carry one truck regardless of the number of orders it is supposed to deliver. The airline has an unlimited number of planes at each airport. However, an airport may only allow a limited number of planes to take off. This available take-off number is denoted by an integer called 'airport capacity' in **airport.csv**. Each airport has its own 'airport capacity'.

Airport Capacity

Each airport is described with the following information in **airport.csv**: airport ID, airport location (x, y) and airport capacity. When a plane departs from an airport, the 'airport capacity' of that airport that **the plane departs from** will reduce by 1. If the 'airport capacity' of an airport becomes 0, no plane can **take off** from that airport. However, other planes can still **land** there.

Example: Given two airports: **[A345, 10.8, -7.3, 5]** and **[A437, -0.4, 8.0, 4]**. For the first airport:

- **A345** refers to its ID.
- **10.8** and **-7.3** refer to its x and y coordinates.
- **5** refers to its initial 'airport capacity'.

Assuming a truck takes a flight from **A345** to **A437**. Before the flight, the airport capacity of the two airports will be 5 and 4 (as stated in the CSV file):

[A345, 10.8, -7.3, **5**] and [A437, -0.4, 8.0, **4**]

After this flight their respective airport capacities will become 4 and 4:

[A345, 10.8, -7.3, **4**] and [A437, -0.4, 8.0, **4**]

The airport capacity of a particular airport will only decrease by one when a plane takes off from that airport. A landing plane will not affect the airport capacity of the airport at which it lands at.

Plane Speed

All the trucks have the same speed as those in q1, and all the planes have the same flying speed which is equal to X times of the truck speed. For simplicity, we denote truck speed as 1 length unit per time unit, and plane speed as X length units per time unit. Both `truck speed' and `plane speed' are floats (not necessarily integers) and given in **parameters.csv**. You can assume that plane speed is always >1.

Here are the CSV files given:

- **orders.csv**. Orders are given in this file. Format: **order ID**, **x-coordinate**, **y-coordinate**.
- **airports.csv**. Information about airports is given in this file. Format: **airport ID**, **xcoordinate**, **y-coordinate**, **initial airport capacity**.
- **parameters.csv**. Format: number of trucks, truck speed (always 1), plane speed (float or integer).

You are expected to create more test cases by editing the values in **parameters.csv** and to "mix and match" the various **orders.csv** and **airports.csv** files.

Requirement Like Q1:

- Your task is to come up with an algorithm that outputs the schedule of each truck.
- The schedules must ensure that all orders are delivered, and
- your algorithm should strive to minimize the time required to deliver all the orders. (i.e. you need to minimize the travelling time of the last truck returning to the company.) However, unlike Q1, your truck may take planes between airports! All distances between locations are calculated in

Euclidean distance; so the distance between the delivery destination for an order and a particular airport will be the Euclidean distance between these two points.

You are required to fill in the body of this function in **q2.py**:

schedule_q2(orders, airports, truck_speed, plane_speed, number_trucks)

where:

- **orders** is a list of items describing the ID and delivery location of each order.
- **airports** is a list of items describing the ID and location of each airport. Each item in this list represents the airport ID, its x-coordinate, its y-coordinate and its airport capacity. This list may look like this:

  [[A345, 10.8, -7.3, 5], [A437, -0.4, 8.0, 4], …]
- **truck_speed** is a float that denotes the speed of trucks (always 1)
- **plane_speed** is a float that denotes the speed of planes (always >1).
- **number_trucks** is a positive integer that denotes the number of available trucks. The function should return a list of schedules in a list of items in which each item describes the schedule of the truck. The list contains the same number of items as the number of trucks used in your schedules. This number may be equal to or smaller than the number of available trucks.

Although usage of airports in your proposed routes is likely to decrease the time taken for the journey, it is not mandatory for your algorithm to use the airports.

Example: The x^{th} item in your returned list may look like this: **[O002, O005, A437, A111, O003, A384, A231, O001]**. This is to be interpreted as:

1) The x^{th} truck starts from the company (0, 0) and goes to the delivery location of order '**O002**' to fulfil the order.
2) It goes to the location of the order '**O005**' to fulfil the order.
3) It goes to airport '**A347**' and
4) then flies to airport ~~'**A384**'~~ '**A111**'
5) It goes to the location of the order '**O003**' to fulfil the order
6) It goes to airport '**A384**' and
7) then flies to airport '**A231**'
8) It goes to the location of the order '**O001**' to fulfil the order.
9) Truck then returns to the company (0, 0).

In this case, the total time taken for the x^{th} truck to complete its route will be the sum of:

1) Distance between the company and '**O002**' **/ truck_speed**
2) Distance between delivery locations of '**O002**' and '**O005**' **/ truck_speed**
3) Distance between delivery locations of '**O005**' and '**A437**' **/ truck_speed**
4) Distance between delivery locations of '**A437**' and '**A111**' **/ plane_speed**
5) Distance between delivery locations of '**A111**' and '**O003**' **/ truck_speed**
6) Distance between delivery locations of '**O003**' and '**A384**' **/ truck_speed**
7) Distance between delivery locations of '**A384**' and '**A231**' **/ plane_speed**
8) Distance between delivery locations of '**A231**' and '**O001**' **/ truck_speed**
9) Distance between delivery locations of '**O001**' and the company) **/ truck_speed**

The time taken by the truck which takes the longest time to return to the company will determine the time taken for all orders to be fulfilled. Remember that your algorithm aims to minimize this time. The number of items in the returned list should be fewer or equal to **number_trucks.**