Final Exam of Algorithms and Data Structures 2022-2023

Consider a highway described as a sequence of **service stations**. Each service station is located at a distance from the start of the highway, expressed in kilometers as a positive integer. There are no service stations which can share the same distance: each service station is uniquely identified by its distance from the start of the highway.

Each service station is equipped with an electric rental car park. Each vehicle is distinguished by its battery range, expressed in kilometers, as a positive integer. The car park of a single station includes up to 512 vehicles. By renting a car from a station, it is possible to reach any other station at a distance that is less than or equal to the vehicle's range.

A journey is defined as a sequence of service stations where a driver stops. It begins at one service station and ends at another, passing through zero or more intermediate stations. It is assumed that the driver cannot turn back during the journey and rents a new car every time they stop at a service station. Therefore, for two consecutive stops *s* and *t*, *t* must always be further from the starting point than *s*, and *s* and *t* must be reachable using one of the available cars at *s*.

The objective of the project is the following: given a pair of stations, plan the route with the **minimum number of stops** between them. In case there are multiple routes with the same number of stops (i.e., tied), the route that favors the stop closest to the start of the highway should be chosen. In other words, let us consider the set of *n* routes ranked by merit $P=\{p_1,p_2,\ldots,p_n\}$, where each route is a tuple of m elements $p_i=(p_{i,1}, p_{i,2},\ldots,p_{i,m})$, which correspond to the distance from the highway's start for each stop. We must choose the only path p_1 such that there is no other path p_j, with the same number *k* of final stops where the previous stop has a smaller distance, i.e., $p_{j,m-k} < p_{i,m-k}$.

Here is an example of a highway. In this example, the correct path between the station at a distance of 20 and the one at 50 is $20 \rightarrow 30 \rightarrow 50$ (and not $20 \rightarrow 45 \rightarrow 50$). Notice that $50 \rightarrow 30 \rightarrow 20$ is also a correct path between the station at a distance of 50 and the one at a distance of 20 (therefore, from right to left).

The input text file contains a sequence of commands, one per line, in the following format. All values, positive integers or zeros, can be encoded in 32 bits.

- **aggiungi-stazione** (add-station) #distance #number_of_cars #car range-1 ... #car range-n
  Adds a station located at the indicated distance, with a fleet of cars, with each car having the specified range.
  Example:
  ```
  aggiungi-stazione 10 3 100 200 300
  ```
  Add a station at a distance of 10 from the highway's start, with a fleet of three vehicles with ranges of 100, 200, and 300 km, respectively. If a station already exists at the indicated distance, the command does nothing.
  Expected output: **aggiunta** (added) or **non aggiunta** (not added).

- **demolisci-stazione** (delete-station) #distance
  Removes the station located at the indicated distance, if it exists.
  Expected output: **demolita** (deleted) or **non demolita** (not deleted).

- **aggiungi-auto** (add-car) #station_distance #car_range_to_add
  If the station exists, ***adds a car*** to it. Multiple cars with the same range can be added.
  Expected output: **aggiunta** (added) or **non aggiunta** (not added).

- **rottama-auto** (delete-car) #station_distance #car_range_to _delete
  Removes a car from the station at the indicated distance, if the station exists and has at least one car with the specified range.
  Expected output: **rottamata** (deleted) or **non rottamata** (deleted).

- **pianifica-percorso** (plan-route) #departure_station_distance #arrival_station_distance
  Requests to plan a route with the restrictions indicated above.
  Expected output: the stops in order, represented by the distance of the stations on the highway, separated by spaces and followed by a line break. The start and destination must be included. If the same station is seen twice, print it only once. If no route exists, print **nessun percorso** (no route). The action of planning a route does not alter the stations or their vehicle fleet. The stations are always considered to be present.