

Algorithms and Data Structures Project 2022-2023

Consider an arbitrarily long highway described as a sequence of service stations. Every station is uniquely identified by its distance from the start of the highway (in kilometres).

Every station has a parking lot that can contain up to 512 electric cars. Every car is identified by its battery capacity, expressed in kilometres.

If you take a car from the station s , you can reach a station t only if the distance between s and t is less or equal to the car battery capacity.

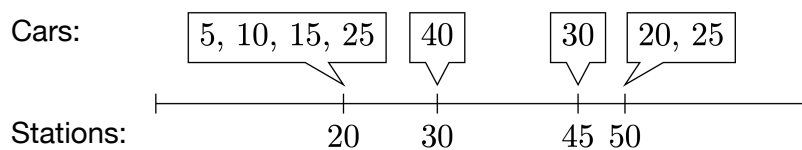
A route is identified by the sequence of stations where the driver stops by to swap cars.

Assume that a driver can't go back during a trip.

The objective is to plan a route between two stations with the minimum number of stops needed to complete it.

If there are more than one different routes with the same minimum number of stops, you must choose the one that stops closer to the start of the highway.

Consider this example:



If you wanted to plan a route from station 20 to station 50, the correct sequence of stations would be [20 30 50]:

From station 20 the driver can take the car with the maximum battery capacity and reach both station 30 and 45, from either one of those stations the driver could reach station 50. However you must prefer stops that are closer to the start of the highway so the possible route [20 45 50] has to be excluded. The route [20 30 45 50] is still possible but has too many stops.

Similarly if you wanted to plan a route from station 50 to station 20, the correct sequence of stops would be [50 30 20].

Input file format and expected output

- **add-station** distance number-of-cars battery-capacity-1 ... battery-capacity-n

Example: **add-station** 10 3 100 200 300

Adds a station 10 kilometres away from the start of the highway, with 3 cars in the parking lot that have a battery capacity of 100, 200 and 300 km respectively. If there was a station at Km 10 already, the command has no effects.

Expected output: **added** or **not added**

- **demolish-station** distance

Removes the station situated at the specified distance, if there is one there.

Expected output: **demolished** or **not demolished**

- **add-car** station-distance car-battery-capacity

Example: **add-car** 100 25

If station 100 exists, a car with a battery capacity of 25 km must be added to its parking lot. There may be 2 different cars with the same battery capacity.

Expected output: **added** or **not added**

- **scrap-car** station-distance car-battery-capacity

Example: **scrap car** 100 25

If station 100 exists, a car with a battery capacity of 25 km must be removed from its parking lot. If such station or car don't exist, it does nothing.

Expected output: **scrapped** or **not scrapped**

- **plan-route** start-station-distance end-station-distance

Example: **plan-route** 20 50

Plans a route from station 20 to 50 according to the previously mentioned specifications. Note that planning a route doesn't affect the position of the cars.

Expected output: the stations where the driver stops to swap cars, including the first and the last station, in the order that they are visited. If the route doesn't exist: **no route**