Coding Exercise

A map is made up of cities and towns connected by highways of different distances. A is connected to B via a road of 100km. B is connected to F via a road of 300km. F is connected to Town E via a road of 50km, and to G via road of 70km. E is connected to G via a road of 150km, and to H via 30km of road, and also to D via 80km of highway. G is connected to H via 50km of road. H is connected to D via 90km of road and D is connected to C via 200km of road, and C is 30 from A directly.

- Create a program that can determine the shortest distance between any places on the map going via the highways.For example if the user gives the program the following json data:

  { "start":"A", "end":"F" }

  the result will be

  { "distance":360 }

- Ensure the user of the program can calculate results when the map data is changed. The program is required to permit addition and deletion of roads as well as destinations. Given

  { "A": {"B":80} }

  as input, the system would change the distance from B to A to be 80km (presumably for a new, better road)

- Using

  { "map": [{ "A": {"I":70, "J":150} }]}

  as input, add the destination I with a distance of 42km from A, and J with a distance of 150km from A. The map should be able to be initialised using something like the following json format:

  {

  "map": [

  {"A": { "B": 100, "C": 30 }},

  {"B": { "F": 300}},

  {"C": { "D": 200}},

  {"D": { "H": 90, "E":80}},

  {"E": { "H": 30, "G":150, "F":50}},

  {"F": { "G":70}},

```
        {"G": { "H":50}}
      ]
    }
```

Please submit the code that generates the results. We expect you to write your own map representation and manipulation code. A json parsing library is permitted if json is the chosen input format, and a command line argument parsing library is also permitted.

Answers can be coded using your choice of the following languages. A functional approach is preferred.

- Ruby
- Scala
- Java
- Haskell
- Clojure