Citymapper Router Challenge

Using some data from Open Street Maps (OSM), we have created a data file representing a graph of walkable streets. You may download the file here:

https://s3-eu-west-1.amazonaws.com/citymapper-assets/citymapper-coding-test-graph.dat

The data file contains a list of OSM ids that are contained in the street graph, followed by a list of bidirectional edges, with distances between them given in meters. The edges represent the walkable streets in the graph.

```
<number of nodes>
<OSM id of node>
...
<OSM id of node>
<number of edges>
<from node OSM id> <to node OSM id> <length in meters>
...
<from node OSM id> <to node OSM id> <length in meters>
```

We would like you to write a short program that takes a graph using this representation as input, and computes the shortest walking distance in meters between two given OSM nodes in the graph (assuming all edges are walkable.) For example

```
./run.sh citymapper-coding-test-graph.dat 876500321 1524235806
2709
```

You may use any language you wish, though we are most familiar with Python, Go, Java, C++ and C. You may only use the standard library from your language of choice, excluding any built-in graph libraries (e.g. you may not use networkx in Python.) Ask us if you have any doubts about this. The code should compile and run on Linux or OS X.

Please provide a short README mentioning how to compile the code, and any design decisions or assumptions you have made.

Please email us back a tarball or zip with a README, all of the code, and a `run.sh` so that we can run your code with `./run.sh <path to graph> <from-osm-id> <to-osm-id>`. If a path exists the program should output the length of the shortest path in meters, and nothing else.

We are looking for clean code that solves the problem, and does not do anything else. If you wish, you may briefly mention any ideas for extensions in the README.