

Figure 1: A visual representation of the road system described in file input1.txt.

Task: Uniform Cost Search

Implement uniform cost search for a program that can find a route between any two cities. Your program will be called find\_route, and will take exactly three commandline arguments, as follows:

find\_route input\_filename origin\_city destination\_city

An example command line is:

find\_route input1.txt Munich Berlin

Argument input\_filename is the name of a text file such as input1.txt, that describes road connections between cities in some part of the world. For example, the road system described by file input1.txt can be visualized in Figure 1 shown above. You can assume that the input file is formatted in the same way as input1.txt: each line contains three items. The last line contains the items "END OF INPUT", and that is how the program can detect that it has reached the end of the file. The other lines of the file contain, in this order, a source city, a destination city, and the length in kilometers of the road connecting directly those two cities. Each city name will be a single word (for example, we will use New\_York instead of New York), consisting of upper and lowercase letters and possibly underscores.

IMPORTANT NOTE: MULTIPLE INPUT FILES WILL BE USED TO GRADE THE ASSIGNMENT, FILE input1.txt IS JUST AN EXAMPLE. YOUR CODE SHOULD WORK WITH ANY INPUT FILE FORMATTED AS SPECIFIED ABOVE.

The program will compute a route between the origin city and the destination city, and will print out both the length of the route and the list of all cities that lie on that route. For example,

find\_route input1.txt Bremen Frankfurt

should have the following:

distance: 455 km

route:

Bremen to Dortmund, 234 km

Dortmund to Frankfurt, 221 km

and

find\_route input1.txt London Frankfurt

should have the following output:

distance: infinity

route:

none

For full credit, you should produce outputs identical in format to the above two examples