

Assignment #1: A Primal Journey

A traveler wants to plan a journey using a map of certain cities, with known roads and distances. Assuming that the trip odometer is set to 0 at the start of the journey, the requirement is that the mileage showing after entering each selected city should be a *prime number* (the traveler considers this an omen of good luck). At the end, the traveler needs to return to the origin city of the journey. Not all cities need to be visited and it is possible that the same city (other than the origin/destination) can be visited several times. The problem is to devise such a “primal” journey.

Input to your program is an edge list representation of the road system: (i, j, d_{ij}) , where d_{ij} is the road distance between cities (i, j) . No ordering of the m input undirected edges (i, j) is to be assumed. You can however suppose that the cities are consecutively numbered $1, 2, \dots, n$. If needed by your code, the values n and m can be input first.

Also provided is a list of the first 125 primes, which should be adequate for your test problems.

Output should indicate the origin city, the sequence of cities visited, and the total mileage for a primal journey. Your code should also report if no such journey exists.

- (1) Run your algorithm on the sample problems (to be supplied) and provide the output, as specified above.
- (2) Express the *space* complexity of your algorithm, in terms of relevant network parameters. You can ignore additive, but not multiplicative, constants.

Things to Keep in Mind:

- make your code as general as possible: applicable to an arbitrary map
- read in the edges one by one; there is no need to store the edge list
- first create an adjacency list representation of the undirected graph G
- then carry out an appropriate search of G , noting that nodes can be visited multiple times
- avoid lots of special cases
- adequately document your code
- try to make your code modular, consisting of reusable routines

In addition to the documented code, provide an English language description of your overall approach, the data structures used, and important program variables. Also discuss any important initializations.