

A program that implements the Dijkstra algorithm on an arbitrary graph, represented as a adjacency matrix or adjacency list.

Input data:

1. An input file with a description of the graph. The input file should have the following form:

n

type

vertex\_1: (vertex, weight) -> (vertex, weight)...

vertex\_2: (vertex, weight) -> (vertex, weight)...

...

vertex\_n: (vertex, weight) -> (vertex, weight)...

Here n is the number of vertices of the graph, type is the type of graph representation in the memory (matrix/list), vertex\_n is a vertex, vertex is the adjacent vertex, weight is the path length between the vertices

2. User-defined parameters:

--start\_vertex - vertex from which the shortest path should be found.

--end\_vertex - the vertex to which you need to find the shortest path.

Basic requirements:

1. Dijkstra's algorithm must work regardless of how the graph is represented in memory.

2. The program must output both the length of the shortest path and the path itself.