

(1)

First of all, we take reading input function and take inputs. It also creates an adjacency list gr.

The algorithm applies Dijkstra's algorithm. It initializes an array `dis` to store the shortest distance. `mn` is used to store pairs (distance, node). It iterates when `mn` is not empty. Then it writes output in `w-out` function. The main part of the code calls `minp` to read values. After the finishing of algorithm, it converts distance into strings. Then it writes output.

(2)

First of all, it calls `minp` to read the input values. It applies Dijkstra algorithm for both source nodes `m` and `n` to find the shortest time. It initializes variable to track the meeting node and minimum time. It compares the maximum of the

two shortest times at each node and updates meet_n and min_t. Finally, It constructs the out tuple with the minimum time and meeting node and calls w-out to write the output.

(3)

First of all, it calls m_inp to read input values and construct adjacency list. It applies Dijkstra algorithm using adjacency list and the total number of nodes. It calls w-out to write the result to the output file.