Task 3

In task01, we had implemented Dijkstra's algorithm where there was a loop used for each edge. As there were M edges, so there were M loops. used for finding the shortest path from the source node to the end node. Moreover, the time complexity for extracting the min value using min-heap and priority queue for M loops is $O(\log N)$.
Overall,

$$\text{Time Complexity} = O((N + M) \log N)$$
$$= O(M \log N)$$

In Task02, as Dijkstra algorithm was used, most of the time complexity remains the same. Except for the part where we modified the code by using the reverse function, the time complexity changes.
Overall,

$$\text{Time complexity} = O(M \log N + R)$$

The algorithm we can use is BFS because it has a time complexity of $O(M + N)$. Furthermore, in BFS, whenever one node is visited, all adjacent nodes are also checked. As a result, minimum number of titans are visited.

Input → BFS (graph, source)