

Task 1

Dijkstra's algorithm is used to find the shortest paths in a weighted graph with present in this task.
from a source to all the other nodes

This algorithm works by iterating through the other nodes until it finds the closest one to the source, then relaxing all the other nodes. The algorithm also produces and utilizes a priority queue to efficiently select the nodes with the minimum distance. The distances are then offloaded to an output file.

Task 2

This code uses a modified version of the Dijkstra's algorithm to find the optimal distance between Alice and Bob, who are the specified starting nodes simultaneously. It determines the minimum time taken for both paths to meet, and identifies the node where they converge. It like the above task, it uses a priority queue to efficiently select the nodes with the minimal distance. The distance and converging node are offloaded to an output file.

TASKS

This code addresses the problem of finding the shortest path through a graph with varying danger levels associated with its edges by implementing a modified Dijkstra's algorithm. It identifies the path with the minimal danger level from the starting to ending node. It also uses a priority queue to efficiently explore the graph, with the resulting danger level being outputted to an output file.