LAB 06

22301645

[NABONITA SAHA]

Task 01 :

Here, ~~to~~ find the shortest path we took the help of Dijkstra algorithm. We at first introduced the formation of a graph. Then from the source node we evaluated the distance to all other nodes, where all other ~~source~~ nodes except source node has a distance of "inf" or infinity. Till all nodes are visited, we find out the ~~#~~ nodes with minimum distance and compare it with ~~a~~ the adjacent ones, the minimum one is updated. Traversing a loop on the list "distances" we generate the output as the "shortest path".

Task 02 :

Here, Dijkstra's algorithm finds the shortest distance from start. We initialize distance of each node as infinity initially other than the start node as 0. We work on a loop untill the priority queue is empty. And for the function meetingpoint() we calculate the distances from friend1 and friend2 to all nodes using Dijkstra algorithm.

Task 03 :

Here, we implement the Dijkstra algorithm to find the maximum danger level path from start to end node. Then it pops the node with maximum danger level from the priority queue and compares the current and previous 'max' danger level and updates it. The main() function works on iterating through the graph and calls the dijkstra algorithm to find the minimum danger level.