16.8.2023

Assignment 6

ID : 19201089
Sec : 9

Task 1

In Dijkstra's algorithm, we find the shortest path from source node to all other nodes in a non-negative graph. Here, there is a priority queue. We take the node with the least distance and update the distances for all of its neighbours such that $dist[U] + cost_{U,V} < dist[V]$ where U = current node, V = neighbour node. This process continues until the queue is empty. The unreable nodes are to be marked with -1 so I have checked for the infinities in the distance array in a while loop and swapped them with -1 until there are none left.

Task 2

Here, for two people - Alice and Bob, we run dijkstra twice from each of their starting node. so, we get two pa arrays with the shortest distances. We compare both array elements in pairs and check if for the minimum of ~~the~~ all the maximums of each pair. This is the time ~~node~~ where the two will meet. However, this check will not be applicable if the node is unreachable by any of the two. The (index+1) is the node where they will meet.

Task 3

Here, we modify the dijkstra algorithm from task 1. Here, we check for $\max(dist[u], cost_{u,v}) < dist[v]$ instead. By doing so, we're getting the max danger for the nodes from the source node. Among these, the result is the danger for the last node.