Q1:

1. Shortest Path Algorithm

Functionn Dijkstra(G, S)

For each vertex V in G

Dist[V] <- infinite

Prev[V] <- NULL

If V != S, add V to Priority Queue Q

Dist[S] <- 0

While Q IS NOT EMPTY

U <- Extract MIN from Q

For each unvisited neighbor V of U

TemperoryDist <- dist[U] + edgeWeight(U, V)

If temperoryDist < dist[V]

Dist[V] <- temperoryDist

Prev[V] <- U

Return dist[], prev[]

1. Minimal Spanning Tree

// Initialize result

Mst\_weight = 0

// Create V single item sets

For each vertex v

Parent[v] = v;

Rank[v] = 0;

Sort all edges into non decreasing

Order by weight w

For each (u, v) taken from the sorted list E

Do if FIND-SET(u) != FIND-SET(v)

Print edge(u, v)

Mst\_weight += weight of edge(u, v)

UNION(u, v)

Q2:

Diagram

Description automatically generated

A whiteboard with writing on it

Description automatically generated with medium confidence

A whiteboard with writing on it

Description automatically generated with low confidence