Task.1

Here I used raw Dijkstra algorithm to find the shortest path starting from a source that is pre determined. I used a priority Que and a distance dictionary. ~~The~~ I appended the nodes serially inside the priority queue Then in the distance dictionary I appended "Infinity" for all nodes.

Then I made the source's distance zero because the distance of the source from the source is alway zero.

Then I relaxed the neighbors to find the minimum path.

Task 2

(I) First I used dijkstra ~~on~~ pointing on Alice's point

(II) Then on Bob's point

(III) Then I created minimun meet point. Where I checked the possibility of the path based on "Infinity". I kept a boolean var "duck-debug" where I retuned 'False' If I couldn't find any valid path.

And true If I found a possible path

task3

Here I implemanted the DSU ~~sto~~ data structure. The initialize function sets up the parents and graph. The find function recursively finds the root. The friend_circle_sizes function processes a list of union query. ~~Ro~~ Then returns the size of the resulting sets after each ~~ea~~ query.

Task 4

Here used prim algorithm to find the mst and calculated the weight of the mst.

I randomly picked my source as the first node. Then I ~~p~~ removed that node from the "Remainingset" and added to the "mst_set".

Then with the while loop I iterated the rest of the remaing nodes. And found their minimum span with the ~~mst_set~~. source.

If my node was disconnected I broke my loop.