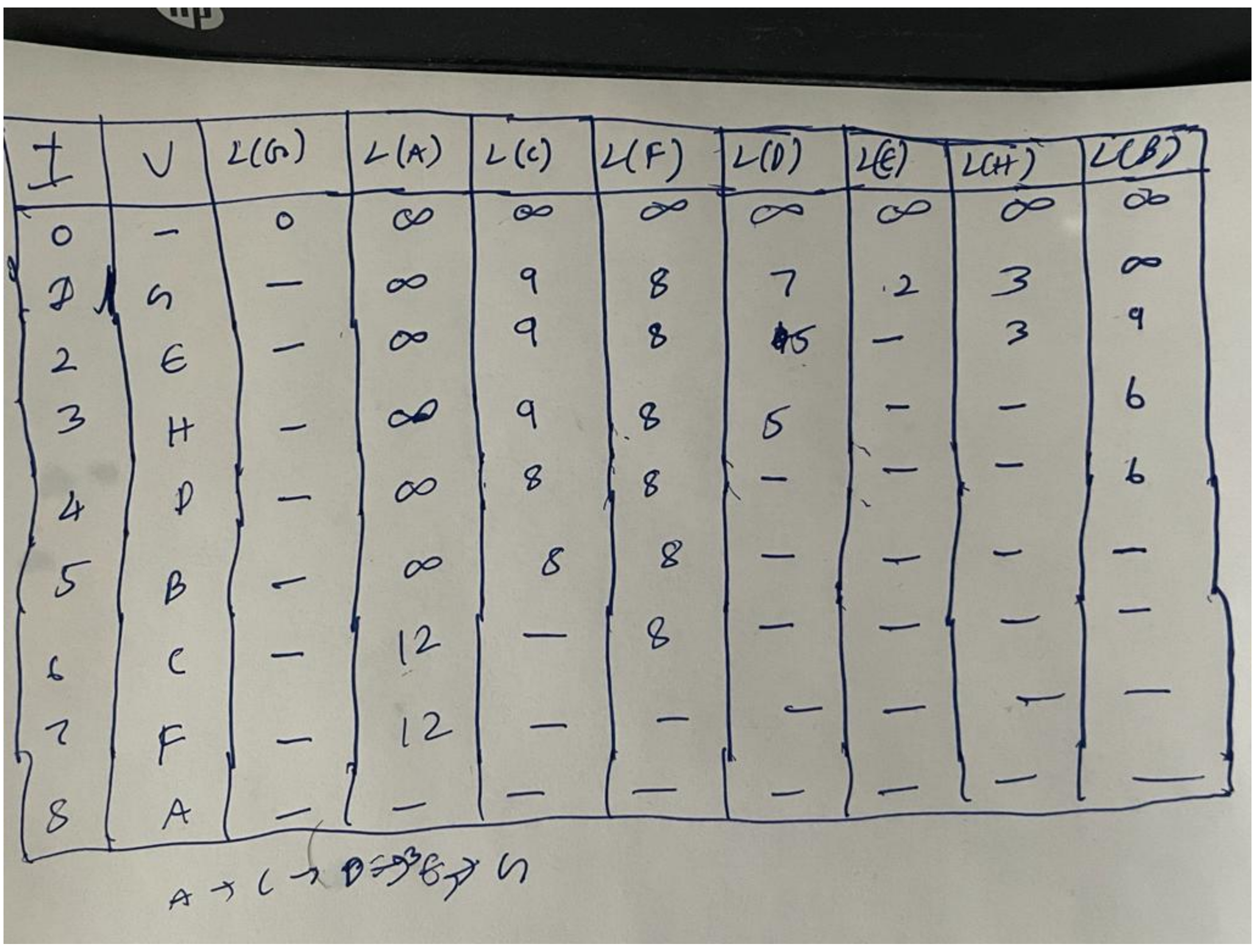
Pranav Prabhu

Problem 1

A. Dijkstra's Algorithm is recommended to find the fastest route from the distribution center to each of the towns. The steps are as follows:

1. Starting from point G and calculate the shortest route to reach adjacent node.
2. Of the nodes that have been updated, move to the node that has not been processed (not already in optimal solution) and has the smallest weight.
3. Add the current node to the optimal solution. From that node, calculate the shortest path to reach adjacent nodes.
4. Repeat steps 2 and 3 until all of the nodes are in the optimal solution.



I	V	L(G)	L(A)	L(C)	L(F)	L(D)	L(E)	L(H)	L(B)
0	—	0	∞	∞	∞	∞	∞	∞	∞
1	G	—	∞	9	8	7	2	3	∞
2	E	—	∞	9	8	~~7~~5	—	3	9
3	H	—	∞	9	8	5	—	—	6
4	D	—	∞	8	8	—	—	—	6
5	B	—	∞	8	8	—	—	—	—
6	C	—	12	—	8	—	—	—	—
7	F	—	12	—	—	—	—	—	—
8	A	—	—	—	—	—	—	—	—

A → C → D → B → G

B. Pseudocode:

To find the first optimal location, we use a for-loop to traverse through t towns.

optimal_location():

```
longest_distance = 0                  // distance from first optimal to farthest town
smallest_dist = 0              // distance from optimal to farthest town in shortest time
Vertex v
for each town t in t towns:
        initialize graph G
        Dijkstra(G, V)
        longest_distance = longest path from optimal town to another town
        if (longest_distance <= smallest_distance): v = optimal town
```

The runtime of the for-loop is O(t) and the runtime of Dijkstra's algorithms is O(t^2). Therefore, the overall runtime is O(t^3).

C. If we were to apply the brute force method to execute Dijkstra's algorithm on every possible vertex, we would find the fasest and "optimal" town (indicating minimal time travelled to the farthest town) to locate the distribution center would be town E.

D. Pseudocode:

To find 2 optimal locations, we add 2 for-loops; however, the second loop excludes the first optimal town which has already been taken. So, we only need to traverse through t - 1 towns; otherwise, both towns will result in the same optimal solution.

optimal_locations():

```
longest_distance = 0                  // distance from first optimal to farthest town
smallest_dist = 0        // distance from first optimal to farthest town in shortest time
for each town t in t towns:
        initialize graph G
        Dijkstra(G, V)
        longest_distance = longest path from optimal town to another town
        if (longest_distance <= smallest_distance): t = first_optimal_town
```

```
longest_distance2 = 0          // distance from second optimal to farthest town in G
smallest_dist2 = 0       // distance from second optimal to farthest town in shortest time
for each town t in t - 1 towns:
        initialize graph G
        Dijkstra(G, V)
        longest_distance2 = longest path from optimal town to another town
        if (longest_distance2 <= smallest_distance2): t = seond_optimal_town
```

Each of the 2 for-loops execute Dijkstra's algorithm. So, runtime is O($2t^3$).

E. If we were to apply the brute force method to execute Dijkstra's algorithm on every possible vertex, we would find the fasest and "optimal" towns (indicating minimal time travelled to the farthest town) to locate the distribution centers would be C and H.