Hussien Jaber
Routing Algorithm Project

Question 1 :

Here I have to find the shortest path between all vertices(Data centers) which are 6 in number. The Floyd warshall algorithm is a perfect choice for this case of scenario of directional graphs. It is really efficient to detect the cost between every vertice pair.

This matrix(Data Structure) shows the shortest Paths, where there is infinity , So that path is not identified:

{[3, 2, 3, 9, 10, 12],
[1, 3, 1, 7, 8, 10],
[INF, INF, 11, 6, 7, 9],
[INF, INF, 5, 11, 1, 3],
[INF, INF, 4, 10, 11, 2],
[INF, INF, 2, 8, 9, 11]}

This matrix can be used as a routing table of each data center. Every array represents the routing table of each data center.

Question 2:

In this problem, I need to find the minimum spanning tree to obtain the shortest path with minimum edges possible to connect with each data center. These edges will represent cables in this case of scenario. As I don't have to consider the directions of edges, and we have a sparse graph which has a lesser number of edges, so the best choice for case is Kruskal's algorithm. Which is based on a greedy algorithm. The data structure used in this algorithm is disjoint-set. Which basically means non-overlapping sets.

Following are the edges in the constructed kruskal's algorithm,:

1 -- 0 == 1
1 -- 2 == 1
3 -- 4 == 1
4 -- 5 == 2
5 -- 2 == 2

Minimum Cost Spanning Tree: 7

If these cables are connected between data centers, It will be the optimum solution in terms of cost of data transmission.

Question 3:

In this problem, I need to find clusters of data centers .These clusters will be defined by the strongly connected nodes in the adjacency matrix. I have found the solution for the question by applying a DFS tree of the forest. From the DFS tree, strongly connected components are found. I have applied Trajan's algorithm to find the clusters between all data centers(vertices).

For the example provided These are the clusters found:
1) 5,4,3,2
2) 1,0

Question 4:

In this problem, I need to use clusters as a single node and create a matrix. There is no specific algorithm in order to solve this problem. In order to create this matrix , I have iterated through all the components in the adjacency matrix. If there is connection, I have checked if it is from the same cluster or different, if the same I didn't care about that, If different I added 1 to my matrix, by adding more edges from one cluster to another, I have created a new matrix of clusters.

For this example provided Matrix is:

0 0
1 0