Title: Generation of a Spanning Tree and addition of links based on Density Gradient.

Problem Statement: Consider a network of 10 nodes. Generate a minimum spanning tree (MST) with the help of Prim's Algorithm. With the MST formed, generate 10 random networks (randomness corresponds to the links added to the MST and the weights of the links) by considering the value of the density gradient to be 2.

Where Nodes -10, Density Gadient- 2, Total Random Network to be formed – 10.

BRIFE THEORY & BACKGROUND:

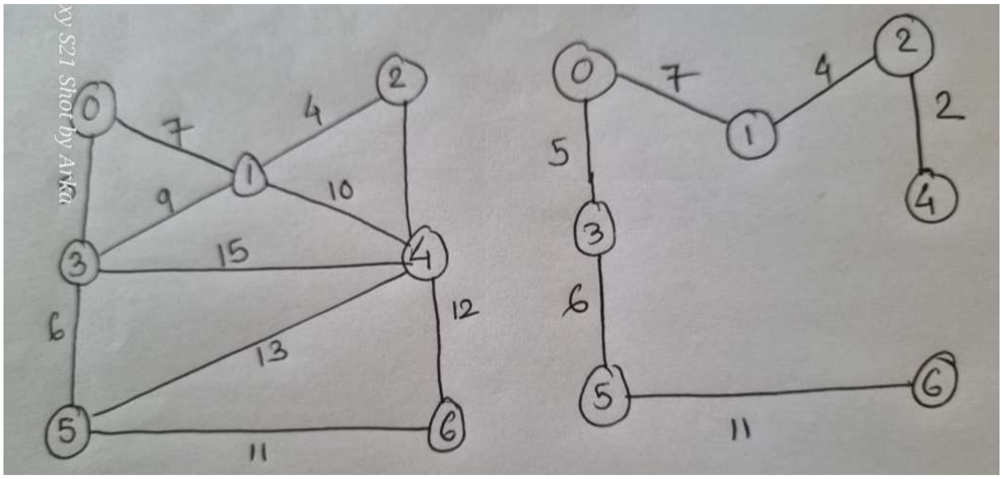
A Minimum Spanning Tree is a fundamental concept in graph theory that plays a crucial role in minimizing the total cost or weight of connecting a set of nodes in a network. Networks are pervasive in various domains, ranging from computer networks to transportation systems. Constructing an optimal network is essential for minimizing resource utilization and ensuring efficient communication. The Minimum Spanning Tree algorithm is a key tool in achieving this goal.

A Minimum Spanning Tree is a subset of the edges of a connected, undirected graph that connects all the vertices together without forming any cycles and has the minimum possible total edge weight. The two primary algorithms used to find a Minimum Spanning Tree are Kruskal's algorithm and Prim's algorithm. "Optimizing Network Efficiency: A Study on Minimum Spanning Trees".

Prims Algorithm:

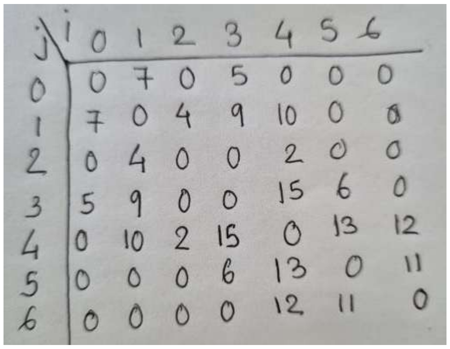
- Prim's algorithm also follows a greedy strategy, starting with an arbitrary vertex and growing the MST by adding the shortest edge that connects a vertex in the MST to a vertex outside the MST.
- It maintains a priority queue or a similar data structure to efficiently select the minimum-weight edge at each step.
- The algorithm continues until all vertices are included in the MST.

As per in lab we are taught by the minimum spanning tree for particular 7 node network. I attached that picture in below. As per the theory number of vertices (V) in the actual network graph and spanning tree will be same. There is the fixed number of edges in the spanning tree which is equal to one less than to vertices (E = V -1). So, if here there is 7 node/vertices then total number of edges will be 6.



So here actually when we are going to extract the minimum spanning tree for the above network it works like- suppose it starts from node 1, then it will search that which edge has min. cost, so it compares and going to 4. When it is at node 4 then again it searching for which vertex to go but here it also checks for if the previous node can have lowest cost, so in prim's algo the main thing it not only check that node only which direction have lowest cost but also all the vertex which it already traverses.

So, the cost adjacency matrix is:



j\i	0	1	2	3	4	5	6
0	0	7	0	5	0	0	0
1	7	0	4	9	10	0	0
2	0	4	0	0	2	0	0
3	5	9	0	0	15	6	0
4	0	10	2	15	0	13	12
5	0	0	0	6	13	0	11
6	0	0	0	0	12	11	0