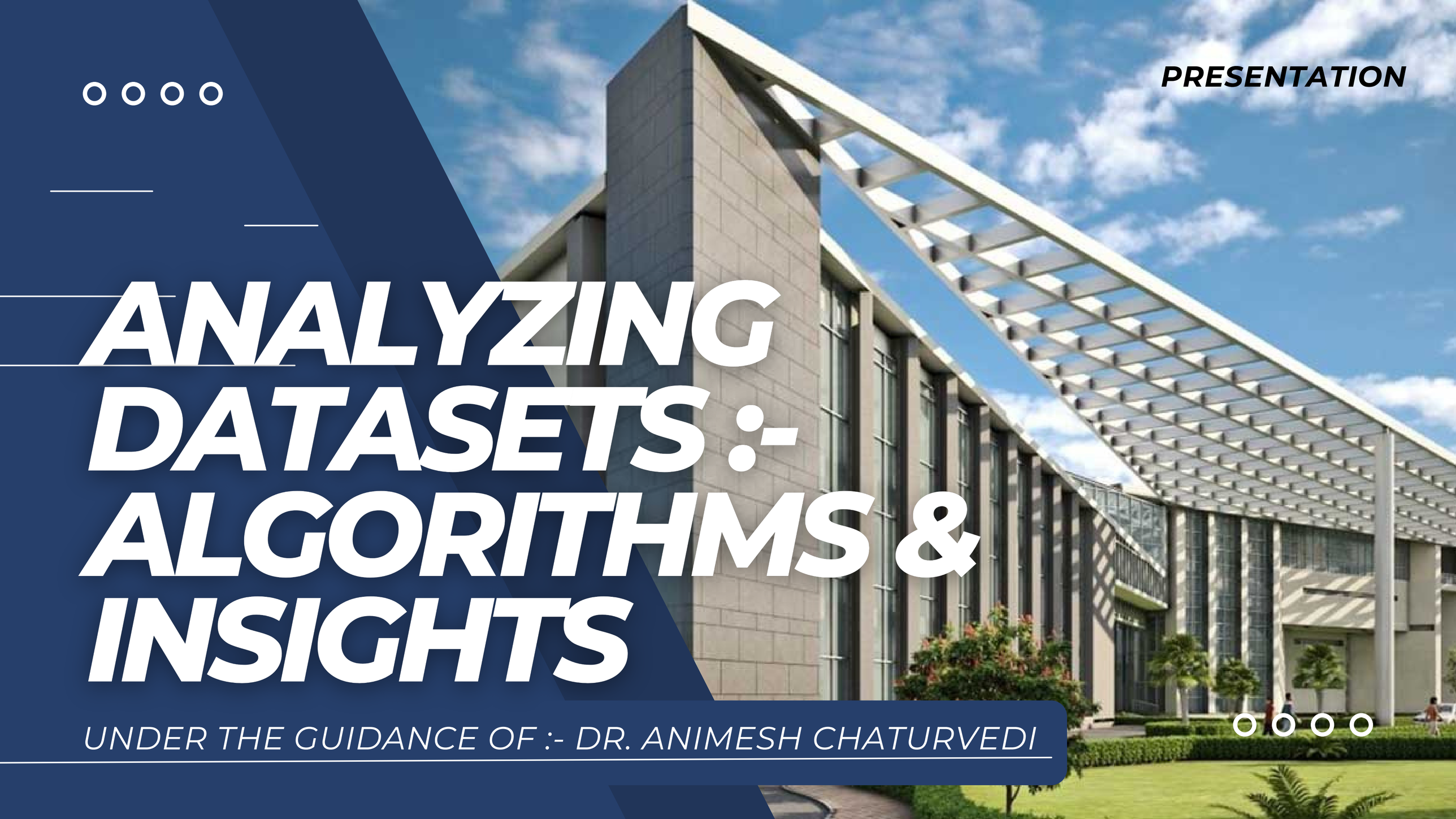
PRESENTATION

ANALYZING DATASETS :- ALGORITHMS & INSIGHTS

UNDER THE GUIDANCE OF :- DR. ANIMESH CHATURVEDI



OBJECTIVES

OBJECTIVES 01

Enhance Individual Understanding

The assignment aims to improve each team member's individual understanding of the research paper and the associated dataset. By reading and comprehending the paper, team members will gain knowledge about the dataset's source, structure, methodology, and findings.

OBJECTIVES 02

Foster Collaboration and Discussion

The assignment encourages collaboration among team members by sharing their insights and perspectives on the research paper. Through collaborative discussions, the team can explore different viewpoints, identify potential research gaps, and brainstorm innovative ideas for further analysis of the dataset.

INTRODUCTION

This presentation provides a thorough analysis of the mean, median, maximum, and minimum values of the datasets, along with graph implementation using MST (Minimum Spanning Tree) and relevant algorithms. The key algorithms employed include Prim's, Kruskal's, Dijkstra's, and Bellman-Ford algorithms. By understanding these algorithms and exploring the central tendency measures, we can gain deeper insights into comparing the three datasets and obtain valuable outcomes.

NETWORK DATASETS

We have selected three Network dataset of three different companies from the following site :-

https://snap.stanford.edu/data/

*All the datasets are undirected, and weight is assumed to be 1 for all the nodes.

TWITCH

The datasets used are Twitch user-user networks of gamers who stream in a specific language.

LAST.FM

A social network of LastFM users which was collected from the public API in March 2020.

RESISTANCE

The dataset contains dynamic face-to-face interaction networks from 62 games of the Resistance game.

TWITCH :-

- Nodes represent users, links represent mutual friendships.
- Features include games played, liked, location, and streaming habits.
- The graph are collected in May 2018, the task is binary node classification to predict explicit language use.
- Nodes = 34,118
- Edges = 4,29,113
- Density = 0.037

LAST.FM :-

- Nodes are LastFM users from Asian countries and edges are mutual follower relationships between them.
- The vertex features are extracted based on the artists liked by the users.
- The graph was collected from the public API in March 2020
- Nodes = 7,624
- Edges = 27,806
- Density = 0.001

R/CO.

RESISTANCE / COMPANY

RESISTANCE :-

- The dataset contains dynamic face-to-face interaction networks from 62 games of the Resistance game.
- Networks were extracted using the ICAF algorithm from discussions, and the DeceptionRank algorithm was applied to detect and characterize group deceptive behavior.
- Nodes = 451
- Edges = 31,26,993
- Networks = 62

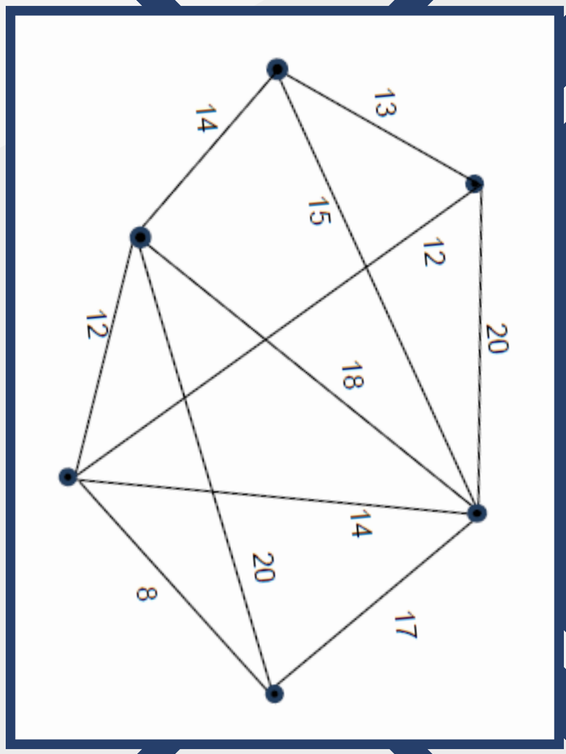
OVERVIEW OF FASCINATING TERM

Before delving into the intricacies of the subject, allow me to provide you with a brief explanation of some terms. These terms will assist you in gaining a clear understanding of the topic at hand.

- MINIMUM SPANNING TREE
- SHORTEST PATH
- PRIM'S
- KRUSKAL'S
- DIJKSTRA'S
- BELLMAN FORD'S

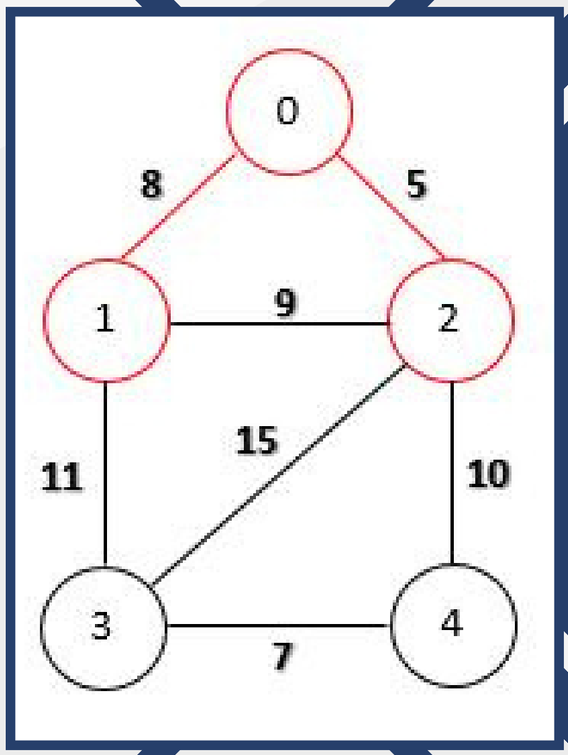
MINIMUM SPANNING TREE

A minimum spanning tree is a fundamental concept in graph theory. It refers to a tree that spans or covers all the vertices of a connected, weighted graph with the least possible total weight. This tree connects all the vertices while minimizing the total sum of the edge weights, ensuring that there are no cycles within the tree. Minimum spanning trees find applications in various fields, such as network design, transportation planning, and clustering algorithms.



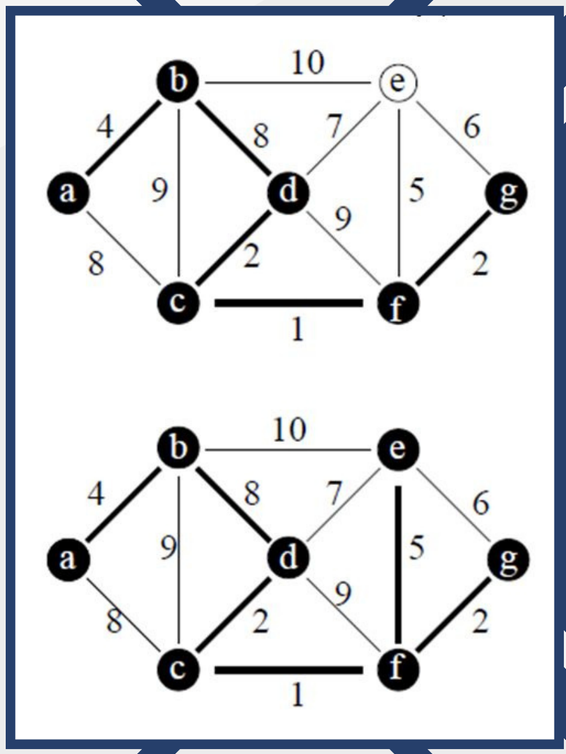
SHORTEST PATH

A spanning tree of an undirected graph G is a connected subgraph that covers all the graph nodes with the minimum possible number of edges. The shortest path (MST) is a variant of the Minimum Spanning Tree concept that focuses on finding the shortest paths between all pairs of vertices in a graph. It aims to identify the tree that connects all vertices with the minimum total path length. This concept finds applications in network optimization, routing algorithms, and transportation planning.



PRIM'S ALGORITHM

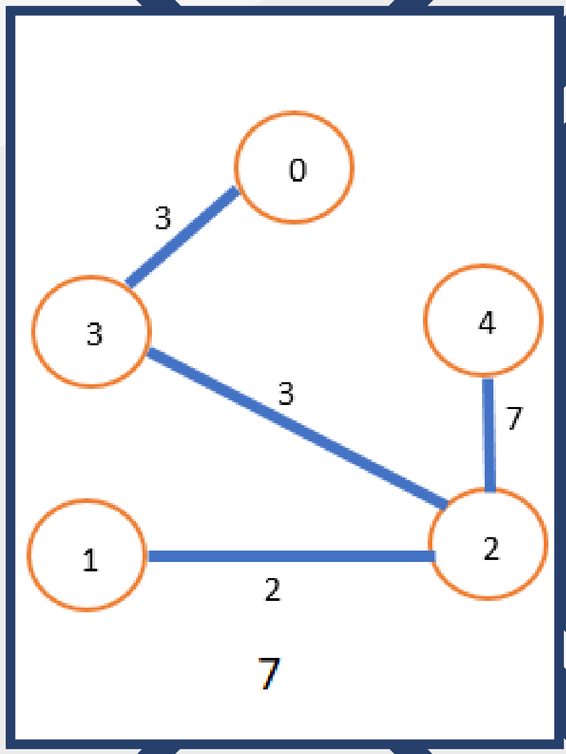
Prim's algorithm is a greedy algorithm used to find the minimum spanning tree (MST) of a weighted graph. It starts with a single vertex and incrementally grows the tree by selecting the edge with the smallest weight that connects a vertex from the existing tree to a vertex outside the tree. This process continues until all vertices are included in the MST. Prim's algorithm ensures that the resulting MST has the minimum total weight among all possible spanning trees of the graph.



KRUSKAL'S ALGORITHM

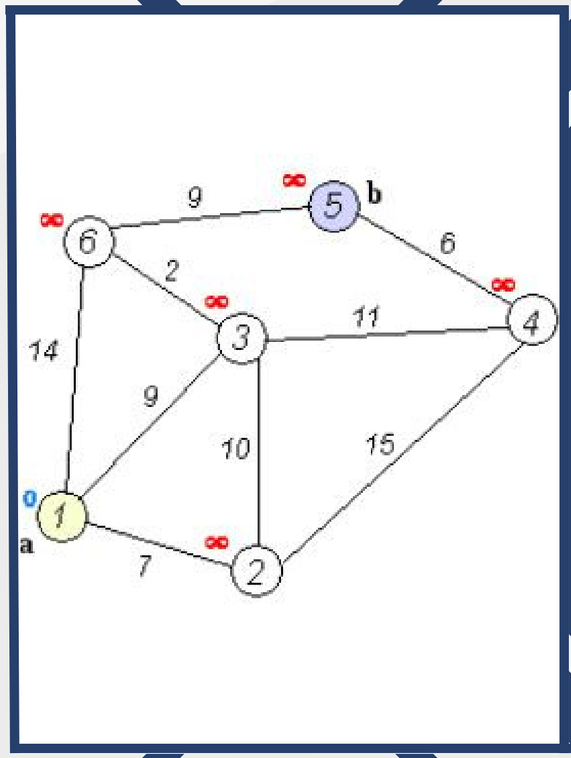
Kruskal's algorithm is a greedy algorithm used to find the minimum spanning tree (MST) of a weighted graph. It starts with each vertex as a separate component and then iteratively merges the components by selecting the edges with the smallest weights.

This process continues until all vertices are connected, forming a tree without any cycles. Kruskal's algorithm guarantees the creation of an MST with the minimum total weight among all possible spanning trees of the graph.



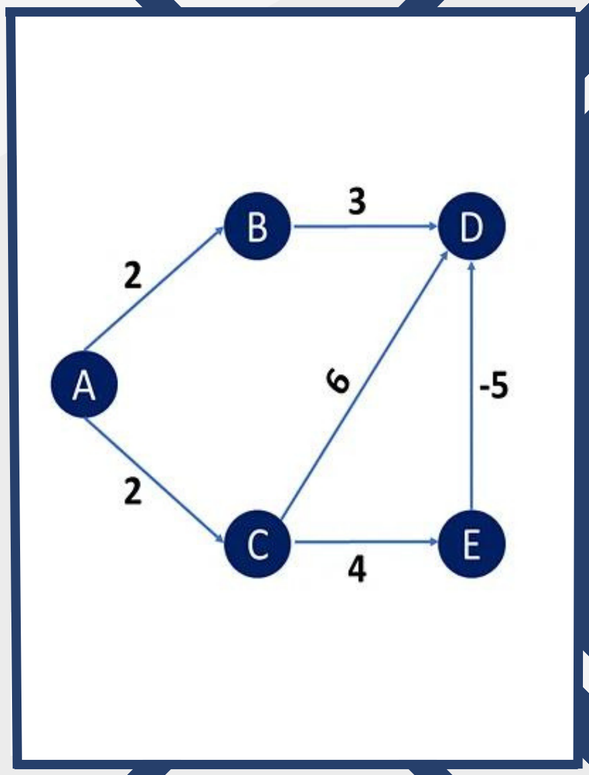
DIJKSTRA'S ALGORITHM

Dijkstra's algorithm is a widely used algorithm for finding the shortest path between a starting vertex and all other vertices in a weighted graph. It works by iteratively selecting the vertex with the smallest tentative distance and updating the distances of its neighboring vertices. This process continues until the shortest path to all vertices is discovered. Dijkstra's algorithm is efficient and guarantees finding the shortest paths in graphs with non-negative edge weights.



BELLMAN-FORD'S ALGORITHM

Bellman-Ford's algorithm is a versatile algorithm used to find the shortest path between a starting vertex and all other vertices in a weighted graph, even in the presence of negative edge weights. It starts by assigning a tentative distance value to every vertex, initially set to infinity except for the starting vertex, which is set to 0. The algorithm then iteratively relaxes the edges, reducing the distance values until the shortest path is found or negative cycles are detected.



GRAPH (MIDSEM)

	PRIM'S	KRUSKAL'S	BELLMAN FORD'S	DIJKSTRA'S
MEAN	11.333	11.333	6.33	6.33
MEDIAN	12	12	6	6
MINIMUM	10	10	4	4
MAXIMUM	12	12	10	10

This chart shows the graph representation of Mid-semester questions.

*For Bellman Ford and Dijkstra's Algorithm, we have taken the path with highest weight from the source node in each graph.

DATASETS

	PRIM'S	KRUSKAL	BELLMAN FORD	DIJKSTRA
MEAN	7202.33	7202.33	6	6
MEDIAN	4383	4383	5	5
MINIMUM	61	61	2	2
MAXIMUM	7623	7623	11	11

This chart shows the graph representation of Datasets.

*For Bellman Ford and Dijkstra's Algorithm, we have taken the path with highest weight from the source node in each graph.

CODE IMPLEMENTATION :-

To begin, we will commence by writing the code for the algorithms discussed earlier. Subsequently, we will proceed to download the dataset from the designated website. Upon completion of the download, we will import the Princeton library in order to make use of the preexisting functions. Next, we will employ if-else statements and a for loop to read inputs from the text file within the dataset. By applying the aforementioned algorithms, we will be able to calculate the desired outcomes, including the mean, median, maximum, and minimum values.

CONCLUSION

- We can see that the datasets have an asymmetric distribution while the graphs have a symmetric distribution by comparing the mean and median.
- We can conclude that the datasets contain a wide distribution of data points whereas the graphs have a more narrow range by comparing the minimum and maximum values.
- We can spot trends and patterns and compare data from many fields by comparing these values across several datasets.

THANK YOU