***GRAPH-THEORY***

* The Project strictly Deals With Simple Directed Graphs
* The Adjacency List Data-Structure is Used For The Graph Representation For Space Efficiency.
* Concepts like friend function, composition and Object-Oriented Concepts are Used in C++.
* Libraries like STL ,stack , queue , set ,iterators … etc have been exploited for the project. C++ Strong Inbuilt Libraries Are Well Equipped.
* Node Numbers are assigned to Nodes .

Eg., Node A <-->Node Number 1…etc

* DMA and Pointers are Used Wherever Necessary. High use of new operators and delete operators for creating objects of classes.

*BASIC GRAPH FUNCTIONS:*

* Counting Indegree, Outdegree , Degree of a Given Node.
* Showing the Incident Edges,Incoming Edges ,Outgoing Edges on to a vertex.
* Functions to store the collection of such edges and return them when asked are implemented.
* Function To Show The Graph , Its Vertices,Edges,Counting number of Vertices and Edges…etc are implemented.
* All These Functions Simply need to peek the Adjacency List Where the Graph is Stored.
* Simple Linear Loops are Adequate For These Simple Functions.

*ALGORITHMS:*

* Implementations of standard algorithms like

1. Kruskal Algorithm On Equivalent Undirected Graph
2. Prim Algorithm On Equivalent Undirected Graph
3. BARUVKA’S Algorithm On Equivalent Undirected Graph
4. DFS
5. BFS
6. Floyd Warshall’s Transitive Closure
7. Floyd Warshall All Pair Shortest Path Algorithm
8. Shortest Path Dijkstra’s Algorithm
9. Shortest Path Bellman Ford Algorithm
10. Shortest Path DAG Algorithm

Object Oriented and Standard C++ Libraries Were Used To Our Advantage For The Implementation. Pseudo Codes Collected from Various Sources.

* Algorithm For Topoplogical Sort
* Algorithm To Check if Directed Cycle Exists in Graph
* Algorithm to Check if Graph is Cyclic
* Counting the number of Connected components in Graph
* Showing All the connected components present in the graph
* Algorithm to check if A path exists between two nodes.