NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES-FAST
KARACHI CAMPUS

COURSE INSTRUCTOR:
SIR SOHAIL AFZAL

CS-2009 DESIGN ANALYSIS AND ALGORITHM

PROJECT NAME:
ALGORITHM IMPLEMENTAION
SECTION: H

GROUP MEMBERS:
SAMEER KHAN (19K-1446)
FAIZ (19K-0247)
SHAH HUSSAIN (19K-1446)

ABSTRACT:

In our project, we will be implementing six different algorithms:

1. Prims
2. Kruskal
3. Dijisktra
4. Bellman-Ford
5. Floyd Warshall
6. Clustering Coefficient (Local Clustering)
7. Boruvka

INTRODUCTION:

We are using 6 different algorithms to find the shortest path between root to the final node. Shortest path algorithms are basically a problem about finding a path between 2 vertices in a graph such that the total sum of the weights of the edges is minimum.
The biggest advantage of using these algorithms is that all the shortest distances between any vertices could be calculated in O (V3), where is the number of vertices in a graph.

Similarly, we are using different algorithms to compare the total cost and total time needed to implement the algorithm

1. **Prims:** Prim's (also known as Jarník's) algorithm is a greedy algorithm that finds a minimum spanning tree for a weighted undirected graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized.

2. **Kruskal** : Kruskal's algorithm is a minimum-spanning-tree algorithm that finds an edge of the least possible weight that connects any two trees in the forest. It is a greedy algorithm in graph theory as it finds a minimum spanning tree for a connected weighted graph adding increasing cost arcs at each step.

3. **Dijkstra**: Dijkstra's algorithm (or Dijkstra's Shortest Path First algorithm, SPF algorithm) is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. ... For a given source node in the graph, the algorithm finds the shortest path between that node and every other.

4. **Bellman-Ford**: The Bellman-Ford algorithm is an algorithm that computes the shortest paths from a single source vertex to all of the other vertices in a weighted digraph.

5. **Floyd Warshall**: is an algorithm for finding shortest paths in a weighted graph with positive or negative edge weights (but with no negative cycles). A single execution of the algorithm will find the lengths of the shortest paths between all pairs of vertices.

6. **Clustering Coefficient**: Triangle counting is a community detection graph algorithm that is used to determine the number of triangles passing through each node in the graph. A triangle is a set of three nodes, where each node has a relationship to all other nodes.

7. **Boruvka**: It is a greedy algorithm for finding a minimum spanning tree in a graph, or a minimum spanning forest in the case of a graph that is not connected.

PROPOSED SYSTEM:

we ask the user input, based on the input and choice of the user, the respective benchmark file is opened after the conditional check of switch case which is applied for 10 different input options.
On the outcome of the Switch case, the respective Algorithm reads the file and starts executing to find the shortest path among different nodes and calculate the total cost from one node to another.

TIME COMPLEXITIES OF ALGORITHMS:

1. Prims: O(VlogV + ElogV) = O(ElogV)
2. Kruskal: ElogE
3. Dijkstra: O((v+e) log v)
O(e log v)
4. Bellmen Ford: *O(VE)*
5. Floyd Warshall: $O(n_3)$
6. Boruvka Algorithm: O(ElogV)

RESULT:

Benchmark	Prims Total cost in Mbps	Kruskal	Dijisktra (Choose any input node e.g. 5)	Bellman-Ford (Choose any input node e.g. 5)	Floyd Warshall Algorithm	Clustering Coefficient (Local Clustering)
Input 10	24.45	24.45	52.8	52.8	453.6	0.6583
Input 20	51.45	51.45	122.25	122.25	2274.9	0.48284
Input 30	87.6	87.6	162.9	162.9	6271.79	0.69809
Input 40	137.1	137.1	344.55	344.55	13504.2	0.77097
Input 50	133.05	133.05	260.4	260.4	15542.09	0.6114
Input 60	201.15	201.15	557.249	557.249	28674.6	0.7103
Input 70	195.749	195.749	475.35	475.35	36364.5	0.68545
Input 80	249.3	249.3	427.35	427.35	50689.5	0.70086
Input 90	287.1	287.1	765.3	765.3	65643.0	0.78583
Input 100	304.5	304.5	693.149	693.149	80038.79	0.69942