ALGO Project

National University of Computer & Emerging Sciences (FAST-NU)

"Graph Analysis"

Submitted to:

Sir Waqas Sheikh (DAA Instructor)

By:

K190326 Om Kumar
K191320 Vinay Kumar

Submission Date:

[20-12-2021]

Graph Analysis

In this project we will be show how different graph algorithm works and will show a general comparison between them. The algorithms which we will be comparing are as under:

- Prim's MST
- Kruskal's MST
- Dijkstra SPA
- Bell-man Ford SPA
- Floyd War shall AP SPA
- Local Clustering Coefficient in GT
- Bourka's MST

1. Abstract:

Invoice from different algorithms with their benchmark were given to us. In this project we analyzed them to present different results of them under different algorithms.

2. Introduction:

Prim's algorithm is a Greedy algorithm. It starts with an empty spanning tree. The idea is to maintain two sets of vertices. The first set contains the vertices already included in the MST, the other set contains the vertices not yet included. At every step, it considers all the edges that connect the two sets, and picks the minimum weight edge from these edges. After picking the edge, it moves the other endpoint of the edge to the set containing MST.

Kruskal's algorithm finds a minimum spanning forest of an undirected edge-weighted graph. If the graph is connected, it finds a minimum spanning tree. (A minimum spanning tree of a connected graph is a subset of the edges that forms a tree that includes every vertex, where the sum of the weights of all the edges in the tree is minimized. For a disconnected graph, a minimum spanning forest is composed of a minimum spanning tree for each connected component.) It is a greedy algorithm in graph theory as in each step it adds the next lowest-weight edge that will not form a cycle to the minimum spanning forest

Dijkstra's Algorithm works on the basis that any sub path B -> D of the shortest path A -> D between vertices A and D is also the shortest path between vertices B and D. Dijkstra used this property in the opposite direction i.e. we overestimate the distance of each vertex from the starting vertex.

Bourka's algorithm begins by finding the minimum-weight edge incident to each vertex of the graph, and adding all of those edges to the forest. Then, it repeats a similar process of finding the minimum-weight edge from each tree constructed so far to a different tree, and adding all of those edges to the forest. Each repetition of this process reduces the number of trees, within each connected component of the graph, to at most half of this former value, so after logarithmically many repetitions the process finishes. When it does, the set of edges it has added forms the minimum spanning forest.

Bellman Ford algorithm works by overestimating the length of the path from the starting vertex to all other vertices. Then it iteratively relaxes those estimates by finding new paths that are shorter than the previously overestimated paths.

Floyd–War shall algorithm is an algorithm for finding shortest paths in a directed weighted graph with positive or negative edge weights. A single execution of the algorithm will find the lengths of shortest paths between all pairs of vertices.

In graph theory, a clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. Evidence suggests that in most real-world networks, and in particular social networks, nodes tend to create tightly knit groups characterized by a relatively high density of ties; this likelihood tends to be greater than the average probability of a tie randomly established between two nodes (Holland and Leinhardt, 1971; Watts and Strogatz, 1998).

3. Proposed System

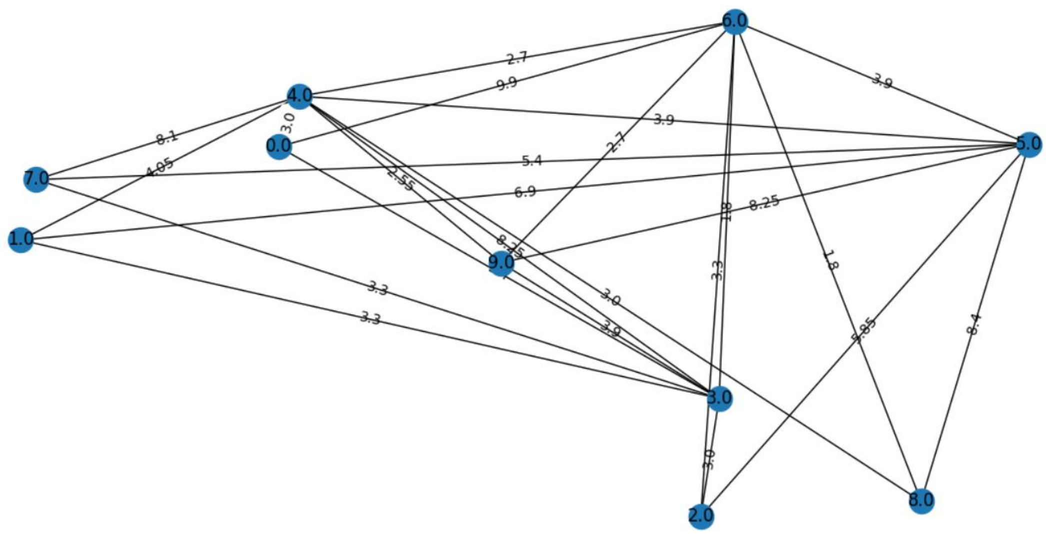
Our system takes input from a file and we are using the python libraries matplotlib, networkx to implement and the graph made by applying the above given algorithms.

4. Experimental Setup

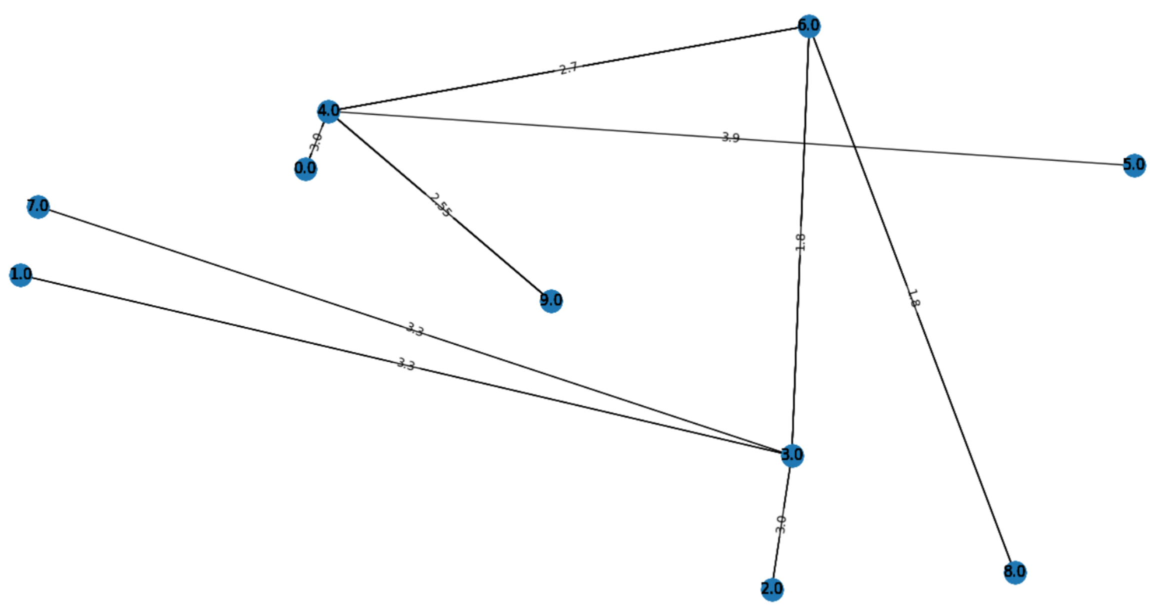
In an operating system we saved parent files containing the data of vertices and edges, which load up in the algorithm through filing. Python 3.10.

5. Result & Discussion

Before Algorithm Applied



After Algorithm Applied



6. Conclusion

Hence we can conclude that our project helps determine the shortest path in different scenarios and this is very helpful in situations when one needs to find which route to take to reach a destination quickly, also it will be very useful for ambulance and many other practical situations. Thus reducing cost and increasing the output, being more effective.

7. Reference

- https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/
- https://en.wikipedia.org/wiki/Prim%27s_algorithm
- https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/
- https://en.wikipedia.org/wiki/Kruskal%27s_algorithm
- https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
- https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/
- https://www.geeksforgeeks.org/boruvkas-algorithm-greedy-algo-9/
- https://en.wikipedia.org/wiki/Bor%C5%AFvka%27s_algorithm
- https://www.geeksforgeeks.org/clustering-coefficient-graph-theory/
- https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/
- https://www.geeksforgeeks.org/floyd-warshall-algorithm-dp-16/