Bharatiya Vidya Bhavan's

SARDAR PATEL INSTITUTE OF TECHNOLOGY

(Autonomous Institute Affiliated to University of Mumbai)

Munshi Nagar, Andheri (W), Mumbai – 400 058.

Experiment No. 6

Aim – Experiment based on graph Algorithms.

Details – The Single-Source Shortest Path (SSSP) problem consists of finding the shortest paths between a given vertex *v* and all other vertices in the graph using Dijkstra's and Bellman–Ford Algorithms.

Dijkstra's algorithm:- Dijkstra algorithm is a single-source shortest path algorithm. Here, single-source means that only one source is given, and we have to find the shortest path from the source to all the nodes.

- The very first step is to mark all nodes as unvisited, mark the picked starting node with a current distance of 0 and the rest nodes with infinity.
- Now, fix the starting node as the current node. For the current node, analyse all of its unvisited neighbours and measure their distances by adding the current distance of the current node to the weight of the edge that connects the neighbour node and current node
- Compare the recently measured distance with the current distance assigned to the neighbouring node and make it as the new current distance of the neighbouring node,
- After that, consider all of the unvisited neighbours of the current node, mark the current node as visited, If the destination node has been marked visited then stop, an algorithm has ended
- Else, choose the unvisited node that is marked with the least distance, fix it as the new current node, and repeat the process again from step 4.

Bellman Ford Algorithm – 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.

- This step initializes distances from the source to all vertices as infinite and distance to the source itself as 0. Create an array dist[] of size |V| with all values as infinite except dist[src] where src is source vertex.
- This step calculates shortest distances. Do following |V|-1 times where |V| is the number of vertices in given graph. Do following for each edge u-v
  - If dist[v] > dist[u] + weight of edge uv, then update dist[v] to
  - dist[v] = dist[u] + weight of edge uv

This step reports if there is a negative weight cycle in the graph. Again traverse every edge and do following for each edge u-v.If dist[v] > dist[u] + weight of edge uv, then "Graph contains negative weight cycle"

Important Links:

1. Reading Resource: Dijkstra Bellman-Ford
2. YouTube Video: Dijkstra Bellman-Ford

Input –

1) A weighted, directed graph G=(V; E) for the case in which all edge weights are non-negative with source s (For Dijkstra's algorithm)
2) A weighted, directed graph in which edge weights may be negative G=(V; E) with source s (for Bellman-Ford)

Output –

1) The Dijkstra Single-Source algorithm returns the shortest paths between a source node and all nodes reachable from the source node.
2) Bellman-Ford algorithm prints a boolean value indicating whether or not there is a negative-weight cycle that is reachable from the source. If there is such a cycle, the algorithm indicates that no solution exists. If there is no cycle, the algorithm prints the shortest paths and their weights.

Submission –

1) C/C++ source code of implementation
2) Verified output for the written source code with multiple inputs
3) One page report of Exp. 6