Single Source shortest Path Problem Write Up:

Theory:

1. The Single-Source Shortest Path (SSSP) problem is a classic algorithmic problem in computer science.
2. It involves finding the shortest path from a given source vertex to all other vertices in a weighted directed graph.
3. There are several algorithms to solve this problem, with Dijkstra's algorithm being one of the most well-known.
4. Dijkstra's algorithm starts with the source vertex and iteratively explores the graph, visiting vertices in an order based on their tentative distances from the source.
5. It maintains a priority queue (often implemented with a min-heap) to efficiently select the vertex with the smallest tentative distance at each step.
6. The algorithm gradually expands from the source vertex, updating the tentative distances of adjacent vertices as it discovers shorter paths.
7. This process continues until all vertices have been visited or the target vertex is reached.

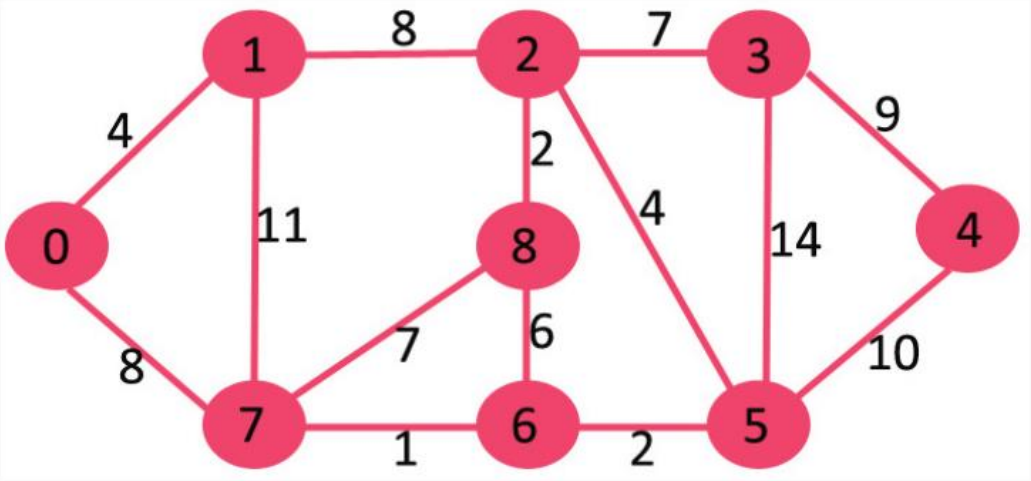
The key properties of single source shortest path problem are as follows:

- Optimality: The SSSP problem aims to find the shortest path from a given source vertex to all other vertices in a graph. The solution guarantees that the distances calculated for each vertex are optimal, meaning that no other path from the source to any vertex has a smaller total weight.
- Weighted Graph: The problem assumes that the graph is weighted, meaning that each edge has a numerical value assigned to it, representing the cost or distance required to traverse that edge. The SSSP algorithms take into account these weights to determine the shortest path.
- Directed or Undirected Graph: The SSSP problem can be applied to both directed and undirected graphs. In a directed graph, edges have a specific direction, while in an undirected graph, edges can be traversed in both directions.
- Non-negative or Negative Weights: Most SSSP algorithms, including Dijkstra's algorithm, are designed for graphs with non-negative edge weights. They cannot handle negative weights because negative cycles can create infinitely decreasing paths. However, there are specialized algorithms like the Bellman-Ford algorithm and the Floyd-Warshall algorithm that can handle negative weights and even detect negative cycles.

Shortest path algorithms have a wide range of applications such as in-

- Google Maps
- Road Networks
- Logistics Research

Input: *src = 0, the graph is shown below.*



Output: *0 4 12 19 21 11 9 8 14*

Explanation: *The distance from 0 to 1 = 4.*

The minimum distance from 0 to 2 = 12. 0->1->2

The minimum distance from 0 to 3 = 19. 0->1->2->3

The minimum distance from 0 to 4 = 21. 0->7->6->5->4

The minimum distance from 0 to 5 = 11. 0->7->6->5

The minimum distance from 0 to 6 = 9. 0->7->6

The minimum distance from 0 to 7 = 8. 0->7

The minimum distance from 0 to 8 = 14. 0->1->2->8

Performance Analysis:

- Time Complexity: The time complexity of Dijkstra's algorithm depends on the implementation of the priority queue data structure. Using a binary heap or Fibonacci heap, the time complexity is typically O((V + E) log V), where V is the number of vertices and E is the number of edges in the graph.
- Space Complexity: The space complexity is O(V) to store the distance values and additional data structures for bookkeeping, such as the priority queue and a visited array.

Additionally, the performance of SSSP algorithms can be affected by graph characteristics, such as the number of vertices, the number of edges, the density of edges, and the distribution of edge weights.

Conclusion:

In this experimental study, we implemented and analyzed the Single-Source Shortest Path (SSSP) problem. Dijkstra's algorithm is efficient for solving the Single-Source Shortest Path (SSSP) problem in graphs with non-negative edge weights, providing optimal shortest paths. Bellman-Ford and Floyd-Warshall algorithms handle negative weights but have higher time complexity, making them suitable for specific cases.