CALIFORNIA STATE UNIVERSITY, FRESNO
DEPARTMENT OF COMPUTER SCIENCE

Class:	**Algorithms & Data Structures**	Semester:	**Fall 2023**

Points		Document author:	**Pratham Aggarwal**
		Author's email:	**Pratham_aggarwal@mail.fresnostate.edu**
		Laboratory number:	**Lab 13**

1. Statement of Objectives

The objective of this lab was to put the Bellman Ford algorithm for determining the shortest distances in a graph into practice. The lab sought to improve understanding and proficiency when working with graph algorithms, with a particular emphasis on the Bellman Ford technique. The scope included taking inputs for the number of vertices, edges, and the source vertex, and displaying the shortest distances from the source to all other vertices. The significance of this lab is in having hands-on experience with graph algorithms. The successful implementation of the algorithm, which provided a clear knowledge of its working principles, was an essential accomplishment.

2. Experimental Procedure

The experiment began by gathering inputs, number of vertices, edges, and the source vertex, as well as the edge list that represented the graph's connections. This input technique allowed for testing with various graph structures. The decision to represent the graph as a simple array of edges in the form [v1, v2, w] helped a simple input process and algorithm implementation.

The Bellman Ford algorithm was implemented according to the coding guidelines provided. The algorithm's starting point was constructed by initializing distance_list to infinity and setting the distance of the source vertex to zero. The iterative process of updating distances V-1 times aimed at refining shortest path estimates by analyzing all possible edges.

During the updating phase, the procedure efficiently handled both positive and negative edge weights, addressing various cases in graph traversal. The implementation of nested loops guaranteed that every edge was taken into account and that the distances were properly updated. The final step was to look for negative weight cycles in the graph to ensure the accuracy of the computed distances.

3. Analysis

Output:

Vertex	**Distance from source vertex**
0	**0**
1	**2**
2	**1**

The analysis of the experiment's output demonstrated that the Bellman Ford algorithm is effective at determining the shortest distances in a graph. The output provided a clear picture of the distances between the source vertex and all other vertices. The program found the shortest pathways while taking into account different graph structures and edge weights.
The output demonstrated the algorithm's adaptability to various graphs. The algorithm's efficiency became clear when it handled graphs with both positive and negative edge weights.

The iterative part in the lab, particularly the V-1 repetitions for relaxation, demonstrated its efficiency in refining distance estimates. This efficiency is essential for large-scale graphs, because searching all possible paths becomes computationally costly. The lab provided insights into the algorithm's time complexity and practical issues when working with graphs of varying sizes by comprehending and applying this iterative process.

Screenshot of the output in the end of the report

4. Encountered Problems

I had difficulty understanding various portions of the coding guidelines when working on the lab, particularly the logic behind specific sections of the code. For example, understanding how to cycle through the edges and update distances, proved difficult. In these cases, I sought assistance from tutors and consulted examples of predefined functions on the internet.

5. Conclusions

This lab gave valuable insights and practical learning experiences by implementing the Bellman Ford algorithm for finding shortest paths in a graph. Understanding the algorithms behind concepts, applying them to real-life problems, and addressing issues that occurred throughout the coding process were all part of the process.
The lab demonstrated the usefulness of graph algorithms. The implementation of the Bellman Ford algorithm provides hands-on experience in traversing graphs and calculating shortest distances by handling concrete instances of edges, vertices, and weights. This experience not only reinforced theoretical knowledge but also improved problem-solving abilities.

6. References

N/A