Title: To find shortest distance from a single source using Bellman Ford Algorithm.

Problem Statement: Write a C program to find the minimum shortest distance from a single source using Bellman Ford Algorithm.

The code should be able to print the shortest distances and the path it traverses. It should also detect the negative cycle if exists.

Comment in the discussion column regarding the Bellman Ford Algorithm with respect to undirected graph with appropriate example.

Brife Theory & Background: The Bellman-Ford algorithm is a dynamic programming-based algorithm used for finding the shortest paths between nodes in a weighted graph. It can handle graphs with negative weight edges and detect negative weight cycles.

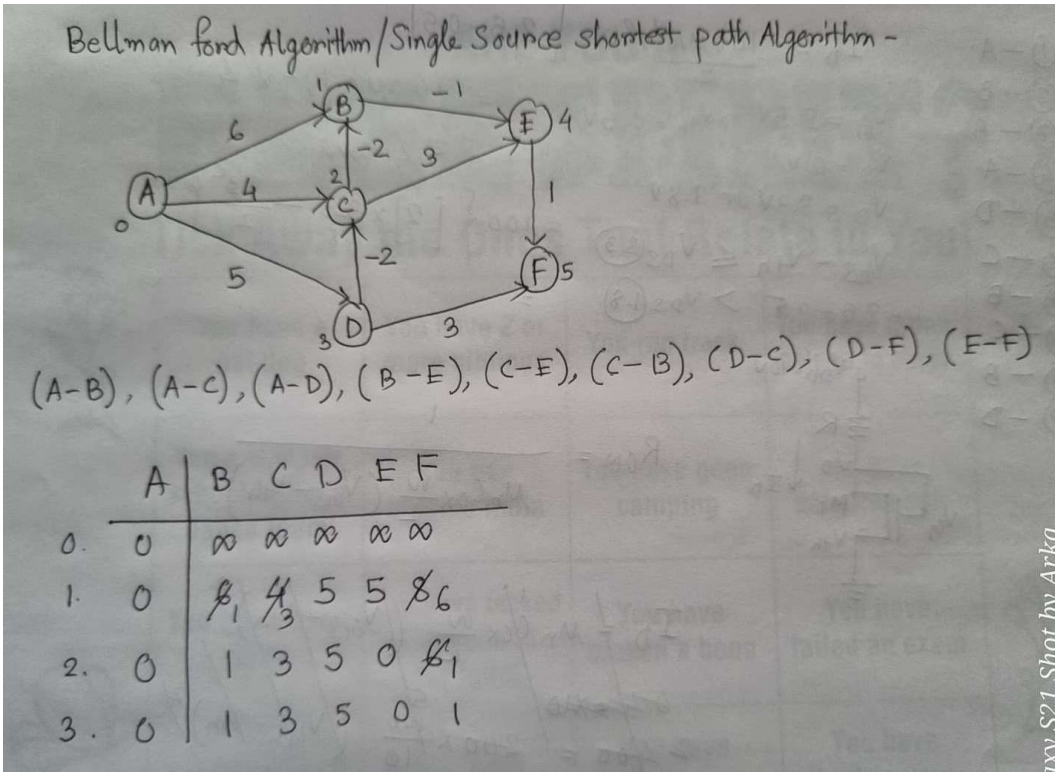
The problem it addresses is finding the shortest paths from a single source vertex to all other vertices in a directed or undirected graph with weighted edges. The weights can represent various metrics, such as distance, time, or cost.

Overview How it works?

1. **Initialization:** Assign an initial distance value to the source vertex as 0 and set all other vertices' distances to infinity.
2. **Relaxation:** Iteratively relax the edges by updating the distance values. For each edge (u, v) in the graph, if the distance to the destination vertex v through the current path (distance[u] + weight(u, v)) is shorter than the previously known distance to v, update the distance[v].
3. **Iteration:** Repeat the relaxation step for V-1 iterations, where V is the number of vertices in the graph. After V-1 iterations, the algorithm guarantees that the shortest paths have been found.
4. **Detection of negative cycle:** After V-1 iterations, check for negative cycles. If any further relaxation is possible, it indicates the presence of a negative cycle in the graph.

The time complexity of the Bellman-Ford algorithm is O(V * E), where V is the number of vertices and E is the number of edges in the graph.

Here I briefly explained with one example which is given in the lab tutorial.



Results: [1.] Code or programming for this network to represent-

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>

#define MAX_VERTICES 100
#define MAX_EDGES 100

// Structure to represent an edge in the graph
struct Edge {
    int src, dest, weight;
};

// Structure to represent the graph
struct Graph {
    int V, E;
    struct Edge edges[MAX_EDGES];
};

// Function to initialize the graph
void initializeGraph(struct Graph *graph, int V, int E) {
    graph->V = V;
```