

108. Bellman Ford Algorithm

AIM: To Find the vertex distance from source by using Bellman Ford Algorithm

PROGRAM:

```
class Graph:

    def __init__(self, vertices):

        self.V = vertices

        self.graph = []

    def add_edge(self, u, v, w):

        self.graph.append([u, v, w])

    def bellman_ford(self, src):

        dist = [float("Inf")] * self.V

        dist[src] = 0

        for _ in range(self.V - 1):

            for u, v, w in self.graph:

                if dist[u] != float("Inf") and dist[u] + w < dist[v]:

                    dist[v] = dist[u] + w

        for u, v, w in self.graph:

            if dist[u] != float("Inf") and dist[u] + w < dist[v]:

                print("Graph contains negative weight cycle")

                return

        print("Vertex Distance from Source:")

        for i in range(self.V):

            print(f"{i}\t\t{dist[i]}")
```

```
g = Graph(5)
```

```
g.add_edge(0, 1, -1)
g.add_edge(0, 2, 4)
g.add_edge(1, 2, 3)
g.add_edge(1, 3, 2)
g.add_edge(1, 4, 2)
g.add_edge(3, 2, 5)
g.add_edge(3, 1, 1)
g.add_edge(4, 3, -3)
```

```
g.bellman_ford(0)
```

```
Vertex Distance from Source:
0          0
1         -1
2          2
3         -2
4          1
```

OUTPUT:

TIME COMPLEXITY: $O(V \cdot E)$