

Problem – 3 Bellman Ford Algorithm

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 print_solution(self, dist):
```

```
    print("Vertex \t Shortest Distance from Source")
```

```
    for i in range(self.V):
```

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

```
def bellman_ford(self, src):
```

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

```
    dist[src] = 0
```

```
    # Relax all edges V-1 times
```

```
    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:
```

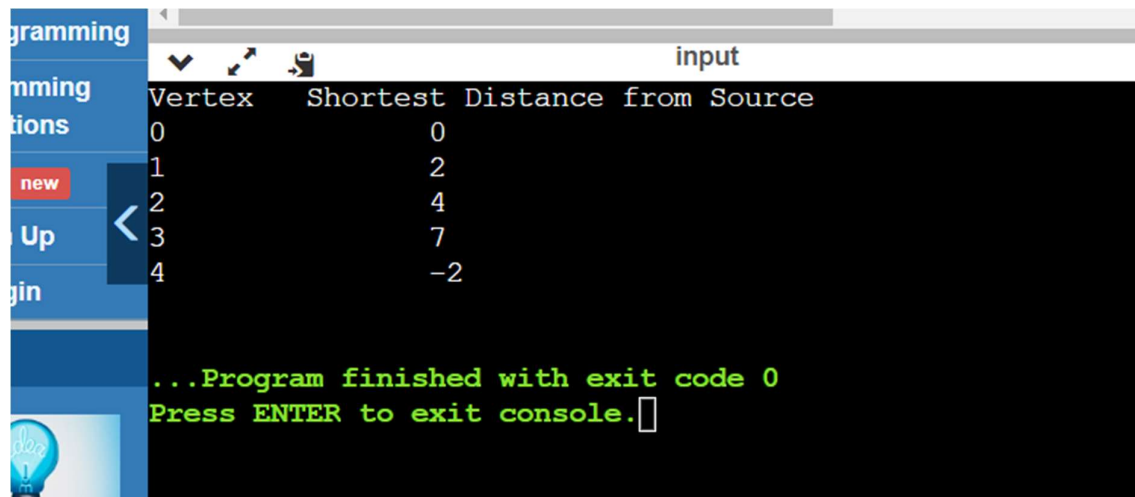
```
        if dist[u] != float("inf") and dist[u] + w < dist[v]:
```

```
            print("Graph contains negative weight cycle")
```

```
            return
```

```
    self.print_solution(dist)
```

```
vertices = 5  
  
graph = Graph(vertices)  
graph.add_edge(0, 1, 6)  
graph.add_edge(0, 3, 7)  
graph.add_edge(1, 2, 5)  
graph.add_edge(1, 3, 8)  
graph.add_edge(1, 4, -4)  
graph.add_edge(2, 1, -2)  
graph.add_edge(3, 2, -3)  
graph.add_edge(3, 4, 9)  
graph.add_edge(4, 0, 2)  
graph.add_edge(4, 2, 7)  
  
source_vertex = 0  
graph.bellman_ford(source_vertex)
```



Vertex	Shortest Distance from Source
0	0
1	2
2	4
3	7
4	-2

...Program finished with exit code 0
Press ENTER to exit console.