EXERCISE-109 Bellman Ford algorithm

PROGRAM

```
def bellman_ford(graph, start):
    distances = {vertex: float('infinity') for vertex in graph}
    distances[start] = 0
    for _ in range(len(graph) - 1):
        for u in graph:
            for v, weight in graph[u]:
                if distances[u] + weight < distances[v]:
                    distances[v] = distances[u] + weight
    for u in graph:
        for v, weight in graph[u]:
            if distances[u] + weight < distances[v]:
                print("Graph contains negative weight cycle")
                return
    return distances
graph = {
    'A': [('B', -1), ('C', 4)],
    'B': [('C', 3), ('D', 2), ('E', 2)],
    'C': [],
    'D': [('B', 1), ('C', 5)],
    'E': [('D', -3)]
}
start_vertex = 'A'
print(bellman_ford(graph, start_vertex))
```

OUTPUT

```
======= RESTART: C:/Users/Gupta/AppData/Loc
{'A': 0, 'B': -1, 'C': 2, 'D': -2, 'E': 1}
```

TIME COMPLEXITY $O(V \cdot E)$