Bellman-Ford algorithm – Problem 5, A3

```
def ford_bellman_algorithm(graph, start_vertex, end_vertex):
    """
     This function creates and uses 2 dictionaries (dist and pred), one
     for retaining the path chosen and one for dist to each vertex
     from the starting vertex
     :param graph: graph object
     :param start_vertex: vertex from graph
     :param end_vertex: vertex from graph
     :return:the lowest cost walk starting from start_vertex to
     end_vertex, or a message if a negative cycle is detected
    """
    dist = {vertex: float("inf") for vertex in graph.inbound.keys()}
    dist[start_vertex] = 0
    pred = {}

    for i in range(len(graph.inbound.keys()) - 1):
        for from_vertex in graph.inbound.keys():
            for to_vertex in graph.outbound[from_vertex]:
                if dist[from_vertex] + graph.costs[(from_vertex, to_vertex)] <
dist[to_vertex]:
                    dist[to_vertex] = dist[from_vertex] +
graph.costs[(from_vertex, to_vertex)]
                    pred[to_vertex] = from_vertex


    for from_vertex in graph.inbound.keys():
        for to_vertex in graph.outbound[from_vertex]:
            if dist[from_vertex] + graph.costs[(from_vertex, to_vertex)] <
dist[to_vertex]:
```

```
            print("Negative cycle detected!")
            return

path = []
current_vertex = end_vertex
while current_vertex != start_vertex:
    path.append(current_vertex)
    current_vertex = pred[current_vertex]
path.append(start_vertex)
path.reverse()
cost = dist[end_vertex]
return path, cost
```