Homework 6 - Problem 2

The algorithm helps find the quickest path from one starting point to every other location, then loops through to find if there is a negative cycle. If so returns “YES”, otherwise returns “NO”

```
negativeCycle:
    Let graph be the graph to check
    Let n be the number of vertices
    Let m be the number of edges
    Let len be a list of size n + 1, to be the length to each location

    for each n:
        for each m: if new len is less than known len:
            Change current len to new len

    for m:
        if new len is still less than known len
            Return “YES” <- Negative Cycle found

    Return “NO”
```

This works because:

The algorithm correctly computes the shortest path from the source to every vertex. Then, the additional loop afterwards checks for negative-weight cycles.

Time Complexity: O(m*n)

The outer loop runs N-1 times, and the inner loop runs M times for each for loop. In the worst case, the algorithm may need to relax all edges for each vertex in each iteration.