**ECE 406**

**Teodor Mihai Tiuca**

**Assignment 3**

1. Done in code
2. Done on paper
3. Done on paper
4. Done on paper
5. Done in code

def getShortestPath(v, prev):

current = v

path = []

while current is not s:

path.append(current)

current = prev[current]

return reversed(path)



The runtime of this algorithm is since the path traced back to s is at most all the other nodes in the graph. This happens if the graph is just a long chain.

1. With the given algorithm, the shortest path from 1 to 3 is 2. This is because the algorithm exits when it reaches node 3. In a regular graph with positive weights, cycles to a node will never be lower cause the node to have a lower weight since all edges add weight. However, in this graph with negative edges, the cycle 1🡪2🡪3🡪1 subtracts the total weight, making the distance from 1 to 1 a total of -1. This means the lowest weight path is impossible to find since repeating the cycle infinitely reduces weight. Therefore, the algorithm’s output is incorrect.