**CS590 homework 6 – Graphs, and Shortest Paths**





**(1. 30pts, 2. 20pts, 3. 25pts, 4. 25pts)**

Remarks:

**Solution to Problem 2:**

To ensure that the generated graph does not contain a negative-weight cycle, you can modify the random\_graph method by checking for potential negative cycles after adding each random edge. If adding a new edge creates a negative cycle, you can simply remove the edge and try another random edge until a valid graph is formed.

Here's a high-level description of the modified approach:

1. Generate a random edge as usual.
2. Check if adding the edge creates a negative-weight cycle using the provided function.
3. If a negative cycle is detected, remove the edge and go back to step 1.
4. Repeat steps 1-3 until the desired number of edges m is reached.

This modification ensures that the generated graph is free from negative-weight cycles. Keep in mind that this approach may result in a longer execution time as it involves additional checks and removals.

**Solution to Problem 3**

Running Time of Bellman-Ford using an adjacency matrix:

The running time of the Bellman-Ford algorithm using an adjacency matrix representation is O(V \* E), where V is the number of vertices and E is the number of edges in the graph. This is because the algorithm performs V-1 passes over all E edges, relaxing each edge in each pass. The negative cycle detection step may require additional passes in case of a negative-weight cycle.