Discrete Event Simulation for Pandemics

Consider an undirected graph G=(V,E) representing a population. Nodes of the graph are individuals. The presence of an edge between two individuals indicates that they are in close contact with each other (e.g, sharing a home or a workplace).

Generate a graph on 100 nodes as follows: for each pair of nodes, toss a fair coin and put an edge between them if and only if you get a heads. This gives you a population graph.

Every individual is in one of the following states: susceptible, infected, or recovered (aka removed). The possible transition for each individual is *susceptible* to *infected* to *recovered*.

- Choose a starting node arbitrarily and call it **s**. At **i=0**, this individual gets infected.
- For **i = 1 to 100**, do
    - For all neighboring nodes of **s** and:
        - If the node is susceptible
            - infect it with probability **0.3**.
                - If it does get infected: the timestamp of infection is **i+1**
                - Generate its recovery timestamp uniformly between **i+2** and **i+7**.
                - **Insert** the infection and recover events into a priority queue--you can use min-heaps, with timestamps as keys. For example, the struct that you use could look like:
                  <TimeStamp: i+5
                  Event type: Infection/Recovery
                  Node id: 5>
    - **s** = **DeleteMin(Queue)**

1. Plot the number of susceptible, infected and recovered individuals, against **i.**
2. Compare the instant at which a node gets infected with it’s shortest distance from the start node s.