COSC 311 Programming Project #1 WINTER 2017

Distributed: 2/7/2017 **Due**: 2/28/2017

Simulation of Single Server versus Two Server Queue
(that is, should your client hire a second cashier?)

Precis: Your job is to quantify the benefit and cost when comparing servicing customers in a single queue with a single server versus servicing customers in a single queue with two servers.

Customers enter a single queue for service. They wait in the queue until they are served or until their wait time in the queue exceeds 8 ticks.

If a customer is assigned to a server, the server is occupied with that customer for however many ticks the customer job will require.

If a customer times out, he is removed from the queue when convenient for your code. Note, no "dead" customer can be assigned to a server.

Benefits:

Each customer serviced adds 1 unit (of benefit)

Costs:

Each server costs 3 units per tick. This is true whether server is active or idle.
A customer that leaves the queue before starting service costs 10 units.

By empirical observation, you estimate that the arrival rate of customers (arrive at the queue) is 1 customer every 5 ticks. Arrival rate is Poisson distributed (see below).

Also through observation, you estimate that the servicing of each customer is uniformly distributed over the half-open range [2, 6) ticks.

The simulation loop is:

```
for each tick {
     remove customer(s) who have had service completed
     update wait times for customers in the queue
     remove customers that have timed-out from the queue
     compute the number of arrivals and insert to the queue
     assign customer(s) to idle server(s)
}
```

At each iteration of the simulation loop, you will need to update relevant statistics such as incrementing each customer's wait time in the queue, the change in total benefit as each customer completes service, the change in total cost as “dead” customers leave the queue.

You will run the simulation twice: once for a single server, once for two servers. The random number sequence must be the same for both simulations.

You will run the two simulations for the same number of ticks (no more than 50 ticks). The number of ticks is chosen by you and is an extent of time that clearly demonstrates that the simulation has reached a stable or oscillating condition or that no new observations and conclusions can be made from the data. You must save the data in arrays and output the arrays after the simulation is completed.

You will need to produce 4 graphs demonstrating the queue’s simulation data (i.e., the results of doing the simulations) versus time.

Graph #1: Give the number of alive customers in the queue at each tick. (ticks on the abscissa, number of customers on the ordinate; two curves)

Graph #2: Give total (accrued) customer wait time versus simulation ticks (ticks on the “x-axis”) for single server and for two servers (two curves)

Graph #3: Give total (accrued) customers serviced versus simulation ticks for single server and for two servers (two curves)

Graph #4: Give accrued benefit data and cost data for single server and for two servers (i.e., 4 curves).

Answer the following questions, explicitly using support from the graphs or from collected data.

(1) Do you recommend adding a second server, and why or why not?
(2) Can a single server keep up with customers entering the system?
(3) Can two servers keep up with customers entering the system?
(4) Justify the number of ticks you used to run the simulation. If you use 50 ticks, explain why that is an acceptable length of time or why it is not an acceptable length of time.

Implementation constraint:

You must use a FIFO queue.

You will need to step through the queue at each interation to update wait times, but the customer will arrive and be serviced in FIFO order.

You must implement a Queue class. You may not import or use an unmodified Queue class from elsewhere. If you cannabilize a Queue class from elsewhere, you must cite the source.

Graphs:

The graphs must be done with software. You can use Excel, Python, MatLab, Mathematica, Sage, etc. You may not use pencil-and-paper. Your simulation programs do not have to produce the graphs, though they may do so if you desire.

Turn in:

- Cover sheet with your name, COSC 311, Project #1 Simulation, URL of code
- Hard copy of Queue
- Hard copy of code for doing the simulations (may be two separate pieces or one piece of code)
- Hard copy of data arrays
- Graphs
- Answered questions.

Poisson: You may use Knuth's algorithm (or use another source) for generating the number of customer arrivals at every tick using arrival rate that is Poisson distributed:
http://stackoverflow.com/questions/9832919/generate-poisson-arrival-in-java

Note, "`mean`" is the average number of customers arriving per tick. So, for 1 customer every 5 ticks, on the average, that means an average of 0.2 customers per tick.