Module 7 Exercises | Joshua Shapiro | 28 May 2017

2. CODE: Download and execute Queue.java, which is a simulation of a single server queue. Cursorily examine the code.
    - What data structures are being used?
    - Where in the code are interarrival times being generated? From what distribution?
    - Where are service times being generated? From what distribution?

- **The data structures are a linkedList and PriorityQueue.**
- **scheduleArrival calls randomInterarrivalTime to generate an interarrival. This random value is being generated by an exponential distribution.**
- **scheduleDeparture calls randomServiceTime to generate a service time. This random value is being generated by an exponential distribution.**

3. CODE: Add code in method randomInterarrivalTime() in Queue.java to estimate the average interarrival time. What does this number have to do with the value of variable arrivalRate in the program?

- **Since arrivalRate is the gamma value, the average interarrival time is 1.3558307457972822, about 1/arrivalRate. The arrivalRate was set to 0.75.**

6. CODE: Execute Queue.java to estimate the average time in system.

- **The average time in system is 2.814388534740915.**

7. CODE: Execute Queue.java to estimate the average waiting time. Subtract this from the estimate of the average system-time. What do you get? Is it what you expect?

- **average system time - average waiting time = 2.814 - 1.891 = 0.923. This means the average service time is 0.923. This makes sense as the service time is pulled from an exponential distribution with a gamma value of 1 (as set by the serviceRate).**

8. CODE: Modify Queue.java to estimate the arrival rate λ and see if it matches the stated arrival rate in the program. Relate this to your earlier estimate of the average interarrival time.

- **The average arrival rate was 1.3473237480392466. This does not match the arrival rate of 0.75, but is extremely close to 1/0.75. This does however match the earlier estimate of the average interarrival time, which does make sense. Effectively, the average time between arrivals should be the same as the average time of arrivals.**

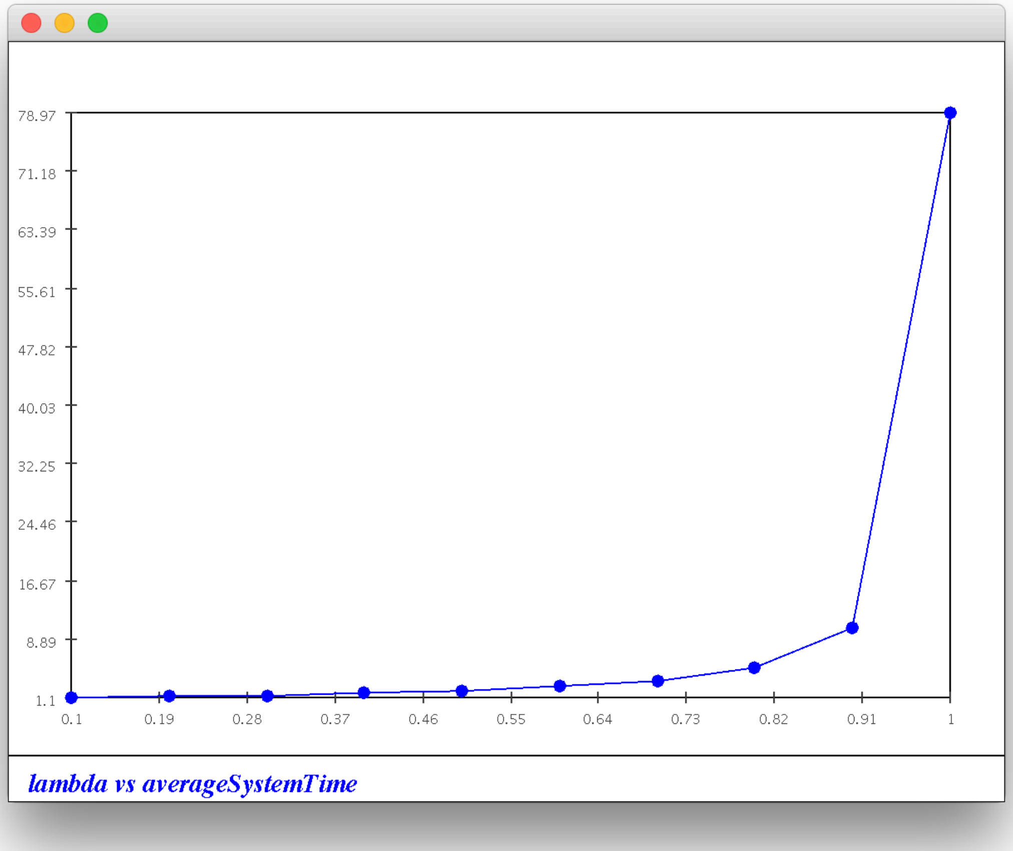
10. CODE: Fix the service rate at $\mu = 1$ and vary the arrival rate: try $\lambda = 0.5, 0.75, 1.25$. What do you observe when $\lambda = 1.25$?

- **As lambda increases, people arrive quicker. This causes longer wait times since the service rate is fixed at 1. When lambda = 1.25, the average wait time becomes 996.55.**

11. What about it?

- **As lambda approaches 0, the number of individuals showing up drastically decreases. This results in an average wait approaching 0, and average system time will match average service time.**

12. CODE: Vary λ as follows: 0.1, 0.2, ... , 0.9 and estimate d. What is the shape of the function d(λ, μ)?



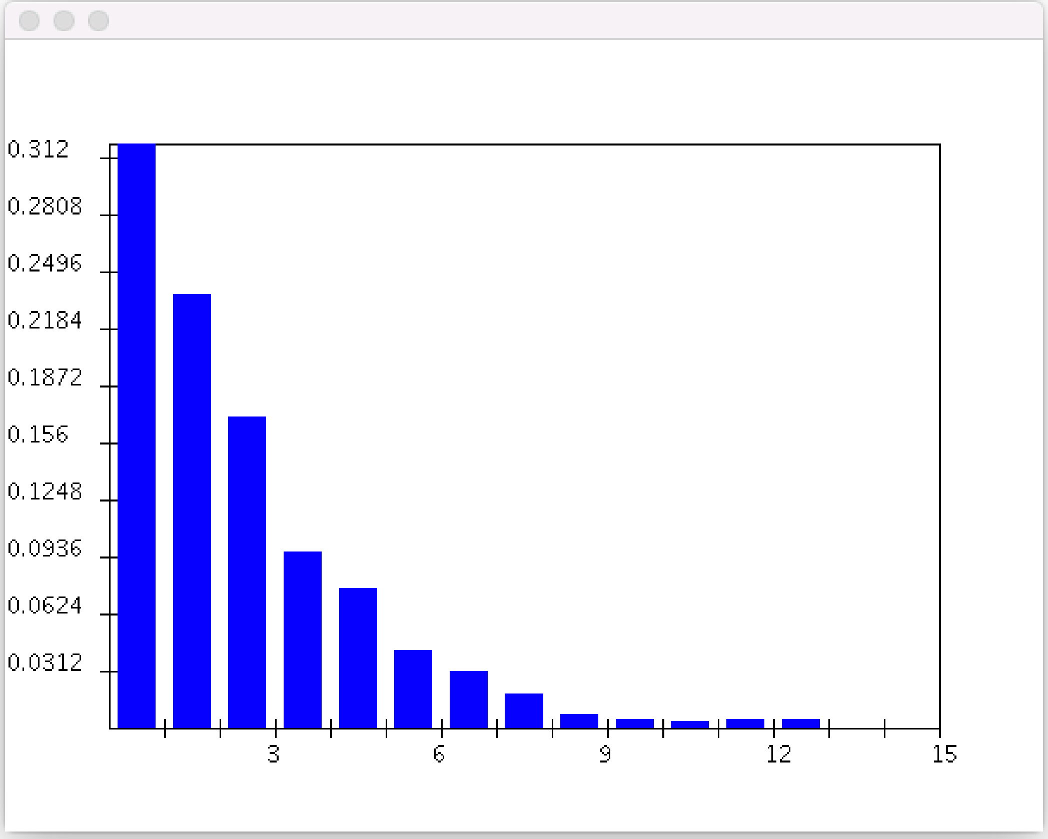
-
- **The function is exponential in nature.**

14. For $\lambda = 0.75$ and $\mu = 1$, estimate the probability that an arriving customer finds the server free. Try this for $\lambda = 0.5$, 0.6 as well. Can you relate this probability to $\lambda = 0.75$ and $\mu = 1$?

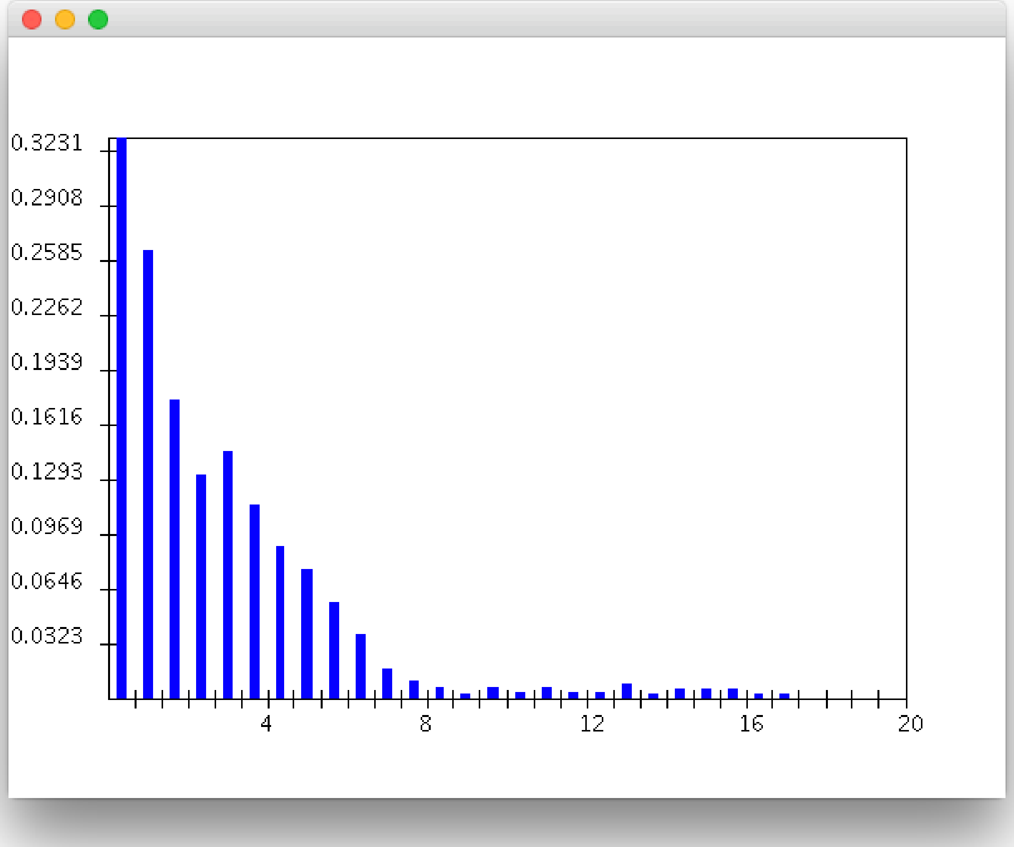
- $\boldsymbol{\lambda = 0.5 \implies 0.499}$
- $\boldsymbol{\lambda = 0.6 \implies 0.4157}$
- $\boldsymbol{\lambda = 0.75 \implies 0.2649}$
- **It seems the probability that an arriving customer finds the server free is 1-λ when $\mu = 1$. See code for how the probability was calculated.**

15. CODE: We will focus on two distributions: the distribution of the number in the system, and the system time. Let rv M denote the number of customers seen by an arriving customer, and let rv D denote the system time experienced by a random customer.
    - Is M discrete or continuous? What about D? What is the range of M? Of D?
    - For $\lambda = 0.75$ and $\mu = 1$, obtain the appropriate histogram of M. Which well-known distribution does this look like?
    - For $\lambda = 0.75$ and $\mu = 1$, obtain the appropriate histogram of D. Which well-known distribution does this look like?
    - Note: use PropHistogram.java or DensityHistogram.java as appropriate.

- **M is discrete since it has to be a positive integer (or 0). It's range is all positive integers and 0.**
- **D is continuous since it can be any positive value (or 0). It's range is [0,∞).**



-
- **Above is the histogram of M. This looks like an exponential distribution.**



-
- **Above is the histogram of D. This looks also like an exponential distribution.**

16. CODE: Download and execute the above simulation: ThreeQueue.java.
    - What is the average system time?
    - Examine the code to see where the system time is being accumulated.
    - Change the queue choice (between 1 and 2) to "shortest-queue" instead of "uniform-random" and see the difference this makes.
    - In what kind of application is "uniform-random" better than "shortest-queue"?
    - Measure the rate of departures from the system and explain the value you get.

- **The average system time is 3.804660969648594.**
- **The system time is being accumulated in the handleDeparture method.**
- **If the shortest queue is chosen, the average system time becomes 3.0558167780687597.**
- **If you want each line to handle the same number of customers (even if one server is slower than another) than uniform random is better.**
- **The average rate of departures would be the time the last customer leaves divided by the number of departures. This equals 0.9900373077350547.**

19. Examine the code in the molecular simulation from Module 4. Is this synchronous or asynchronous?

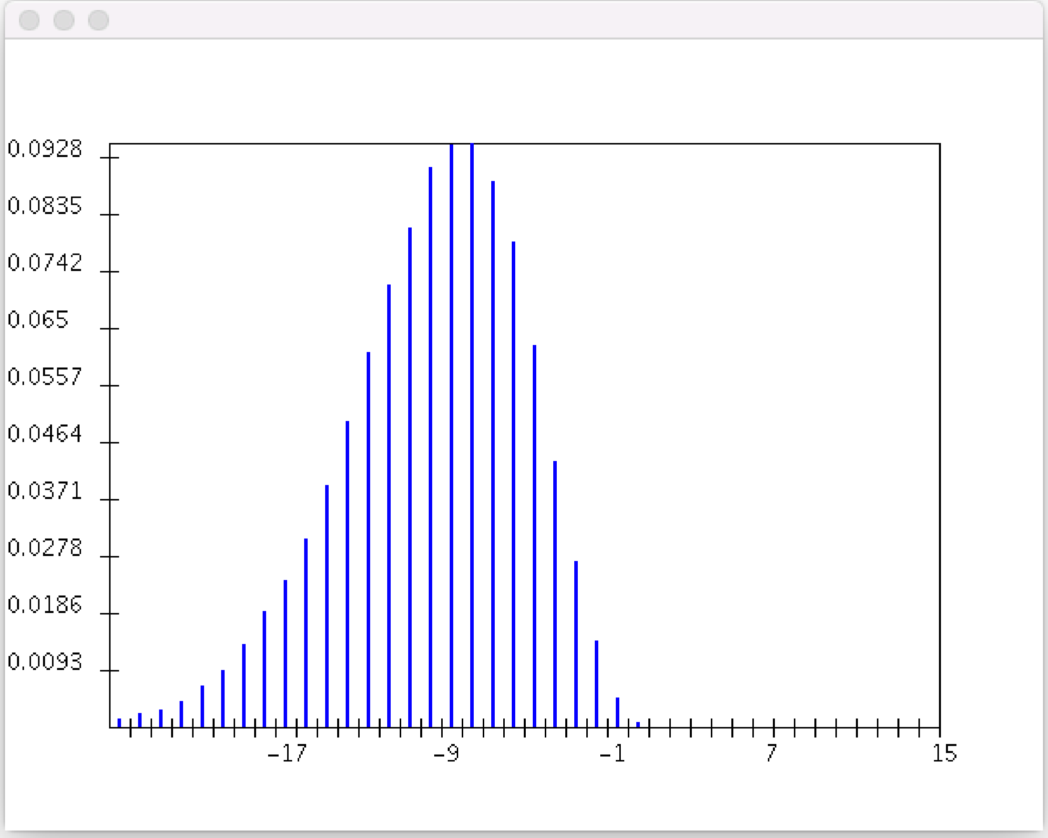
- **This code is synchronous.**

20. Find a simulation of the Game-of-Life. Is this synchronous or asynchronous?

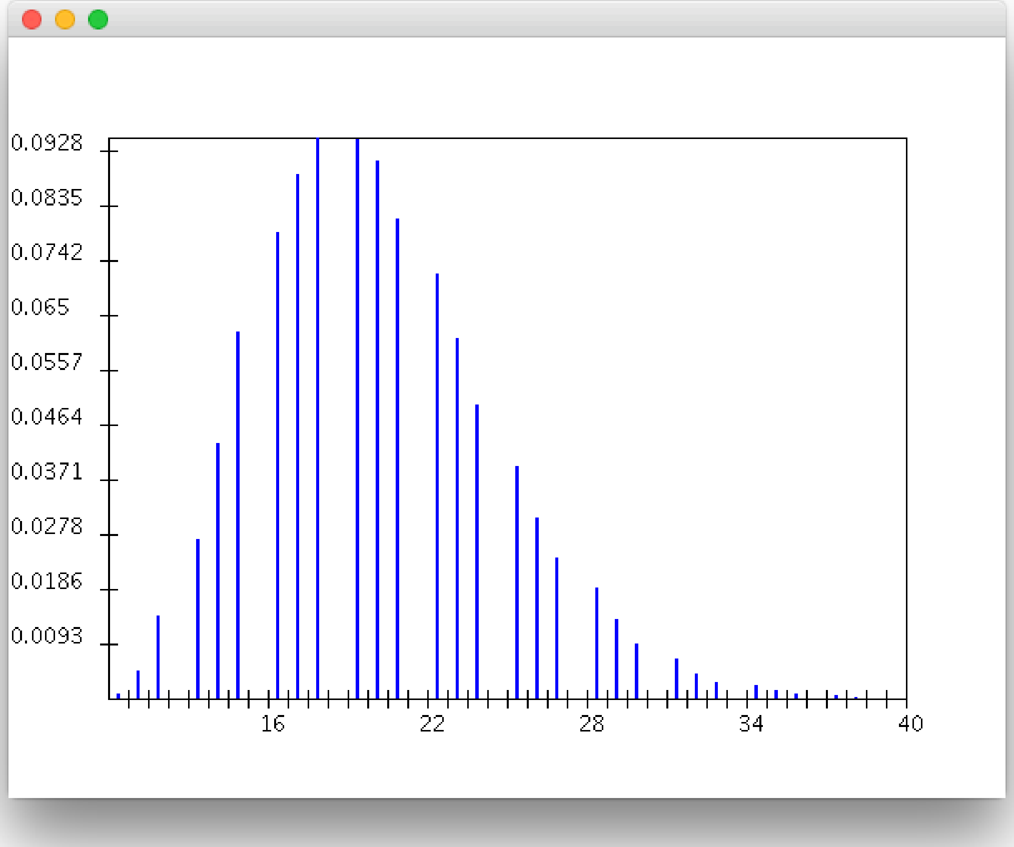
- **At each step everything updates at once, so it is synchronous.**

21. CODE: Fill in code in Raindrop.java to obtain histograms of X and T respectively. Use s = 1, h = 10, p = 0.5.
    - What is the likely distribution of X?
    - Vary h: try h = 20, 30, 40, 50. What is the relationship between E[T] and h?
    - Note: use PropHistogram.java or DensityHistogram.java as appropriate

- **The distribution of both X and T are gaussian (x is the first figure).**



-



-
- **E[T] is about twice the height.**

22. Do you know the historical significance of the distributions of X and T?

- **No, is it that most natural phenomena follow normal distributions?**

23. What is the size of the eventlist for the single-server queue? For the three-queue example?

- **The size of the eventlist is the number of arrivals and departures for the single-server queue. It is the same for the three queue example, though it could be rewritten to use 3 different event queues instead of 1.**

24. If the eventlist has n events, how long does it take for each operation (in order-notation)?

- **If a linked list is being used...**
  - **Removing the first element takes O(1)**
  - **Inserting an event takes O(n)**

25. If instead of a sorted-list, we use an unsorted list, how long does each operation take?

- **If a linked list is being used...**
  - **Removing the minimum takes O(n)**
  - **Inserting an event takes O(1)**