COSC 223: PROBABILITY AND COMPUTING
PROJECT 2: QUEUEING

Due Friday, April 12, 11:59pm

1 Find a Partner: Due Thursday, March 28, 5:00pm

This is a partner project: you will work with one other student in the class, and you may choose your partner. You must work with a *different* partner than the person with whom you worked on Project 1. Your first task is to find a partner and let me know with whom you will be working. By **Thursday, March 28, 5:00pm** you and your partner must submit a single text file (one per team) containing both of your names. For example, if I decided to work with Prof. Alfeld on this project, he and I would submit one text file containing:

```
Kristy Gardner
Scott Alfeld
```

You can submit this through the CS department submission site: `https://www.cs.amherst.edu/submit`, or, from `remus/romulus`, using the `cssumbit` command:

```
cssubmit myteam.txt
```

And then use the menu to navigate to this class and assignment.

If you haven't found someone to work with by Thursday evening, please submit a text file with just your name and I will find a partner for you.

2 Intellectual Responsibility

You will be working with a partner on this project and all aspects of the project can (and should) be done together. Do not share code or written work with anyone besides your partner, and do not look on the internet for any solutions. You may discuss the concepts involved in this project with other students in the class who are not your partner; if you do, please note at the top of your writeup with whom you consulted, and what you discussed.

3 Submission

There are three different components to submit for this project (see Section 4 for details):

1. All code that you wrote for this project
2. A `README` file explaining how your code works and how I can run it
3. A "research paper" writeup describing your experiments and results

You'll submit all of these files in a SINGLE submission by uploading them via the CS department submission web site: `https://www.cs.amherst.edu/submit`. This assignment is due Friday, April 12 by 11:59pm.

4 Your Tasks

In this project, you will study how variability affects performance in a single-server queueing system. In the course of this assignment, you will:

- Implement an event-driven simulator of a single-server queueing system
- Develop insights about how variability affects performance in queueing systems
- Design your own research question and experiments to answer it
- Practice writing a "research paper" describing your experiments and findings

4.1 Implementation and Experiments

The purpose of your experiments is to answer the following research question:

How does job size variability affect mean response time in a single-server queueing system?

To answer this question, you will need to write a program that simulates a single-server queueing system. You will use this simulator to measure mean response time under different assumptions about job sizes. You may use whatever programming language you and your partner prefer.

4.1.1 System Parameters

The system we'll study is parameterized as follows:

- Interarrival times are Exponentially distributed with rate λ, where λ ranges from 0 to 1.
- Service times (job sizes) are Hyperexponentially distributed with mean size $\mathbf{E}[S] = 1$. (Note that the mean service rate is thus $\mu = 1/\mathbf{E}[S] = 1$, so $\lambda < \mu$ and the system is stable.)

  A *Hyperexponential* distribution is a mixture of Exponentials. In this case, we will consider a two-phase Hyperexponential, denoted H_2. It is parameterized by two different rates and a probability. Specifically, if $X \sim H_2(\mu_1, \mu_2, p)$, then

$$X \sim \begin{cases} \text{Exp}(\mu_1) & \text{w.p. } p \\ \text{Exp}(\mu_2) & \text{w.p. } 1-p \end{cases}$$

  That is, with probability p X is drawn from an Exponential with rate μ_1, and with probability $1-p$ X is drawn from an Exponential with rate μ_2.

  From this, you should be able to compute $\mathbf{E}[X]$, $\mathbf{E}[X^2]$, and $\mathbf{Var}(X)$. Our goal is to understand how the variance of job size affects mean response time, so we will let $\mathbf{Var}(X)$ take on different values: 1, 10, 20, and 50.

What parameters should you use for your Hyperexponential? From $\mathbf{E}[X] = 1$ and $\mathbf{Var}(X) = 1, 10, 20, 50$, we have two equations that can help us solve for μ_1, μ_2, and p. But because we have three parameters, we need one more equation to be able to uniquely determine the three parameters. We will add one more restriction to our Hyperexponential: we will only consider Hyperexponentials that have "balanced means," which means that we will set

$$\frac{p}{\mu_1} = \frac{1-p}{\mu_2}.$$

(Do you see why we use the term "balanced means" to describe this property?) Now we can find the values of μ_1, μ_2, and p to achieve each of our desired variances.

In your writeup: Include the computations you did to find $\mathbf{E}[X]$ and $\mathbf{Var}(X)$, as well as a table of the parameters you used for each variance you considered.

4.1.2 Simulation

Your simulator should be structured as an event-driven simulation, as we'll discuss in class on Wednesday. The general structure is as follows:

- Keep track of the current system time, the next time at which a job will arrive, and the next time at which a job will depart.
- Repeat until n jobs have arrived:
  - If the next arrival time is earlier than the next departure time:
    - Advance the current system time to the next arrival time
    - Generate a new job and add it to the queue (or have it enter service, if the server is idle)
    - Update the next arrival time and (if necessary) the next departure time
  - Otherwise (i.e., if the next departure time is earlier than the next arrival time):
    - Advance the current system time to the next departure time
    - "Complete" the current job by removing it from the server
    - If the queue is nonempty, have a new job enter service
    - Update the next departure time

4.1.3 Experiments

Once you've written your simulator, you are ready to use it to study your research question. Write some code that uses your simulator to measure mean response time for different values of λ and different values of $\mathbf{Var}(X)$ (as described above). As with the caching project, when you run your experiments you'll want to ignore any transient effects that occur at the very start of your simulation. To do this, run your simulation for a very long sequence of arrivals (on the order of 10^6) and throw out the first 10^4 or so of them.

Once you've run your experiments, plot your results with λ on the x-axis, ranging from 0 to 1, and $\mathbf{E}[T]$ on the y-axis. You should include four lines, one for each of the four variances you'll test.

4.2 Extensions

In this section of the project, you will explore a second research question about performance in queueing systems. This time, you get to decide what research question to consider. You should be creative, but not overly ambitious given the timeframe of the project. Here are some ideas for questions you might study:

- Suppose that instead of changing the job size variability, we had Exponentially distributed job sizes with mean 1 and we changed the variability of the interarrival times. How does this affect mean response time?
- Suppose we had two servers instead of one, where each server has its own dedicated queue. How should we decide to which server to send an arriving job? One option is to flip a coin; with probability p send the job to one server and with probability $1 - p$ send the job to the other server. Can you come up with a policy that achieves lower mean response time?
- Suppose that instead of FCFS scheduling, your server could use some other scheduling policy. Can you come up with a policy that achieves lower mean response time than FCFS?

You are welcome to use any of these or make up your own. If you decide to make up your own research question, please *run it by me* so I can help you determine if you've chosen an appropriately sized question.

5 Deliverables

You'll submit three things for this assignment:

1. All of the code that you've written for this project
2. A `README` file explaining how your code works and how I can run it. The main purpose of this document is to enable me to use your code. You should explain what command to type to run your code (and what, if any, command line parameters are needed) and explain the input, output, and purpose of each method. You do not need to describe line-by-line what your code is doing.
3. A writeup of your experiments, results, and what you've learned about queueing from your results. The goal of your writeup is to provide a complete description of what you did in your project, why you did it, and what you found. This should be structured similarly to the research paper you wrote for project 1. In project 1, I gave you a detailed outline of what sections to include in your writeup and what content belongs in each section. Now that you've had experience writing this sort of paper, it's your turn to come up with the format and content that will go into the paper. You might want to refer back to project 1 as you're thinking about what to include in this paper and how to organize it.