15-857/47-774 Homework 1: Queueing Terminology & Probability Review

Instructions: Homework is due at the *start* of Friday's class. You have a full week. We grade your homework right away, so please don't be late. If you're having problems, please go to office hours. Feel free to collaborate with other students, but you should write up your own solutions. It is good form to list the names of people with whom you collaborate. Before you begin, it's a good idea to make sure you're totally comfortable with the first 9 chapters of the "Introduction to Probability for Computing" book by Harchol-Balter. Those chapters will teach you *short* ways to solve problems, so that you don't get bogged down in computations.

These problems are from your textbook, *Performance Modeling and Design of Computer Systems*. Starred problems are *not* in your textbook, but are given below. Also, some tips are given below to help you!

Exercises: 2.2, 2.3, 2.4*, 2.5*, 3.24, 3.68*, 4.8*, 4.9*

[Some tips on Exercise 2.3(a)]

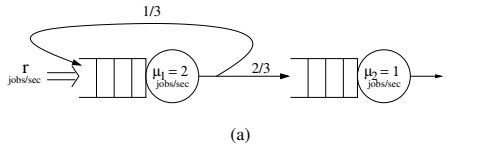
- The scheduling policies considered are preemptible, meaning that jobs can be started, stopped, and resumed where they left off. Thus, at any point of time there may be jobs that are partially completed, even though only one job is run at a time.
- If you are trying to prove that SRPT is *not* optimal, you will need to find a counter-example, i.e., an arrival sequence where SRPT does not produce the best performance. An arrival sequence $\mathcal{A}$ is a sequence:

  $$\mathcal{A} = \{(a_1, s_1), (a_2, s_2), ..., (a_n, s_n)\},$$

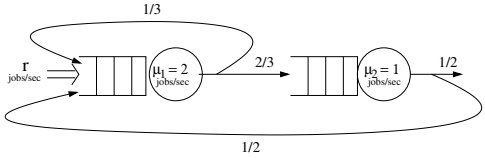
  where a_i is arrival time of the ith job and s_i is the size of the ith job.
- If you are trying to prove that SRPT *is* optimal, then you'll want to argue that SRPT is optimal for *every* arrival sequence $\mathcal{A}$. A popular strategy is proof-by-contradiction. That is, assume there is some arrival sequence $\mathcal{A}$, where $OPT(\mathcal{A})$ is the optimal schedule for $\mathcal{A}$, and assume that the mean response time under $SRPT(\mathcal{A})$ is strictly worse than that under $OPT(\mathcal{A})$. You now want to find a contradiction ... namely you want to prove that $OPT(\mathcal{A})$ can be strictly improved upon.
- In deriving mean response time, you need to take into account the response time of every job. Feel free to assume a finite-length arrival sequence.

Exercise 2.4: [Maximum outside arrival rate]
The figure below shows two queueing networks with probabilistic routing.

(a) For network (a), what is the maximum outside arrival rate r?



(a)



(b)

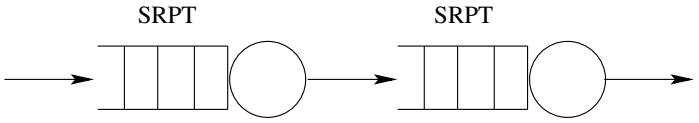
(b) How does the maximum possible r change for network (b)?

Exercise 2.5 [Optimal scheduling for tandem queue]
In a *tandem queue system*, jobs enter at queue 1, and after service they proceed to queue 2. the *response time* of a job is the time from when it first enters queue 1 until it completes service at server 2.

We assume that each job has some known (s_1, s_2) pair, where s_1 denotes the job's service requirement at server 1 and s_2 denotes its service requirement at server 2. Assume that the pair for each job is chosen by some adversary, where s_2 is not necessarily the same as s_1, and different jobs may have completely different pairs.

Our goal is to find a scheduling policy at each queue that minimizes $\mathbf{E}[T]$, the mean response time across all jobs. To simplify the problem, we assume that there are only n jobs, and *all of these are present at time* 0.

Ashley proposes that to minimize $\mathbf{E}[T]$, we should schedule jobs at queue 1 in SRPT order according to their s_1 value. Likewise, we should schedule jobs at queue 2 in SRPT order according to their s_2 value. She calls this the $SRPT^2$ algorithm.



Is $SRPT^2$ optimal? Specifically let $\mathcal{S}$ denote a set of n jobs, all present at time 0. Let $OPT(\mathcal{S})$ denote the mean response time under the optimal schedule for $\mathcal{S}$, while $SRPT^2(\mathcal{S})$ is the mean response time under $SRPT^2$. Is it the case that

$$\forall \mathcal{S},\ SRPT^2(\mathcal{S}) = OPT(\mathcal{S})?$$

If $SRPT^2$ is optimal, then prove it. If $SRPT^2$ is not optimal, can you show that $SRPT^2$ has some approximation ratio $r > 1$? Specifically, can you show that:

$$\forall \mathcal{S},\ \frac{SRPT^2(\mathcal{S})}{OPT(\mathcal{S})} \leq r?$$

How low can you make r? Can you make $r < 2$?

[Note: This is an **optional** open problem. Spend as much or as little time as you want on it. It will only be graded if you get somewhere meaningful on it, in which case I will follow up with you.]

Exercise 3.68 [Staggered installation]
Disk 1 is installed at time 0. Disk 2 is installed at time 10. The disks have lifetimes which are i.i.d. with distribution Exp(λ). What is the probability that disk 2 fails before disk 1? Check your answer on the corner cases of $\lambda \to 0$ and $\lambda \to \infty$.

Exercise 4.8 [Simulation of M/G/1]
In this problem, you will simulate an M/G/1 queue. Do not worry; you do not need to know what this notation means – everything you need to know is explained in this problem. Use any programming language you like. Feel free to work in teams of two for this problem. Only one of you needs to program, but you should both help with figuring everything out.

Your queue is a FCFS queue. The job sizes are drawn i.i.d. from a distribution S with density function

$$f_S(t) = \frac{1}{t \ln(5)}, \quad \text{where } 2 \leq t \leq 10.$$

The interarrival times of jobs are i.i.d. instances of $I \sim \text{Exp}(\lambda)$. You will need to set λ appropriately so that the load of your queue is: $\rho = 0.5; 0.6; 0.7; 0.8; 0.9$.

Your goal will be to determine the mean response time, $\mathbf{E}[T]$, and the mean number of jobs, $\mathbf{E}[N]$, for your queue, under each of the above loads. To get instances of the job sizes and interarrival times, you will use the inverse transform method.

Please submit a copy of your code, and report the following:

(a) What values of λ did you use to get the different values of load.

(b) In generating instances of S, what function did you use to map an instance of Uniform$(0, 1)$ to the job size?

(c) What are your values for $\mathbf{E}[T]$ and $\mathbf{E}[N]$ under each load? Now plot your values of $\mathbf{E}[N]$ as a function of ρ. What trend do you see? Similarly, plot your values of $\mathbf{E}[T]$ as a function of ρ. What trend do you see?

Some notes about running a simulation: We recommend running an event-driven simulation. At every moment of time, you need to be aware of the number of jobs in the system and the time until the "next event." Here an "event" is an arrival or a departure. You should NOT try to generate all job sizes and interarrival times in advance. Instead generate these as you need them. Whenever a job arrives, you immediately generate the time of the next arrival. Whenever a job completes service, you generate the size of the job about to start. Maintain a global clock and always move the clock to the "next event."

Chapter 14 of the book, "Introduction to Probability for Computing" explains this in detail. We are also very happy to talk about this in office hours.

[Hint: If you do this right, you should find that $\mathbf{E}[N] = \lambda \cdot \mathbf{E}[T]$. Do you see that? If not, try running your simulation a lot longer (best to average over at least a million jobs) or averaging over more runs.]

Exercise 4.9 [More practice generating random variables for simulation]
Let X be a continuous random variable with the following density function

$$f_X(t) = \begin{cases} \frac{1}{16}(t-2)^2 + \frac{1}{6}, & \text{if } 0 < t \leq 4, \\ 0, & \text{otherwise.} \end{cases}$$

Provide an algorithm that generates X. (You can assume that you can generate instances of Uniform$(0, 1)$.)