Assignment 2 : CSL 201
Topic : Discrete Event Simulation of a Queueing Network

In this assignment, we will implement a simulator for a queueing network and use it to compute the average time spent by a consumer in the system. During the simulation, we will use the binary heap data structure to store the set of possible next events. A list of test cases for your simulator along with correct outputs is given in the file **test_cases**, a link to which is given on moodle.

1 Exponential Distribution

We say that a random variable X is exponentially distributed with rate λ if and only if:

$$Pr\{X \leq t\} = 1 - e^{-\lambda t}$$

Or, equivalent, the probability that $X = t$ is given by:

$$Pr\{X = t\} = \lambda e^{-\lambda t} \ for \ t \geq 0$$

The expected value of X is $\frac{1}{\lambda}$.

We can generate random numbers which obey the exponential distribution with rate λ using the following code snippet:

```
#include <stdlib.h>
#include <math.h>

float generate_exponential(float lambda) {
float p = ((1.0*rand())/RAND_MAX);
return (1/rate) * log(1/(1-p));
}
```

Here **rand()** is the in-built random number generator in C++. To use this function, we need to include the header files **math.h** and **stdlib.h** as shown above.

The code works as follows. Function **rand()** gives a random positive integer in the range [**0, RAND_MAX**]. We convert it to a random number between 0 and 1 by dividing the output of **rand()** by **RAND_MAX**. By the above formula, the value of the exponential random variable x is given by the equation:

$$1 - e^{-\lambda x} = p$$

Solving this for x, gives $\frac{1}{\lambda} \ln(\frac{1}{1-p})$, which is the value returned by our function.

2 A Single-Server Queueing System

We will now consider the simplest queueing network - one which consists of a single server (see Figure 1).

An example system which can be modeled by this network is a line of customers at a bank counter. There are two parameters λ and μ, which correspond to the customer arrival and teller service rates. These parameters stipulate the following:

1. The inter-arrival time between any two consecutive customers is exponentially distributed with rate λ.
2. The time taken to service a particular customer is exponentially distributed with rate μ.

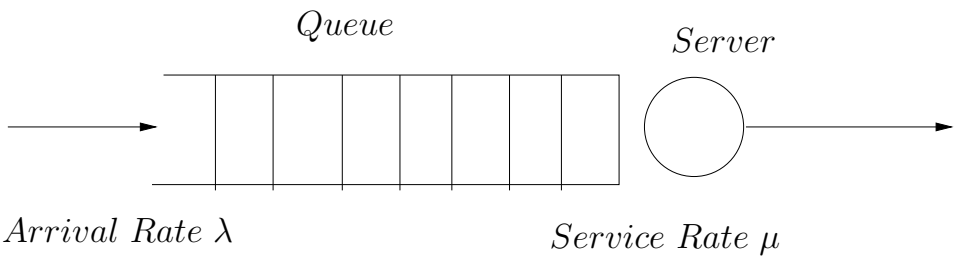

Figure 1: A Single-Server Queueing System

Therefore, on average, λ customers arrive per unit of time, and μ customers are served by the teller in one unit time. We assume that all queues are first-in first-out i.e., customers are served in the order in which they arrive.

A major quantity of interest in any queueuing system is the average time spent by a customer between entering and exiting the network. We denote this quantity by the symbol W.

Let us qualitatively describe the behaviour of this system. If the arrival rate λ is faster than service rate μ, the queue of customers will grow indefinitely and the average time spent by a customer in the system can become arbitrarily large. Thus, a steady-state of this system can be reached only when $\lambda < \mu$ i.e., when service rate is at least as large as arrival rate. In this case, one can show the average time spent by a customer in the system approaches a limit as the time progresses.

For example, it can be shown that for the above one queue system, $W = \frac{1}{\mu - \lambda}$.

3 Queueing Networks

The problem becomes more complex when the output of one queue is routed to another queue. Figure 2 shows a sequential network of two queues. The

customer arrival rate is 5 at input Queue 0. All customers queue up at Queue 1, which has a service rate of 6. A customer served at Queue 1 joins the waiting line at Queue 2, which has a service rate of 7. Finally, customers leave the system after being served at Queue 2.

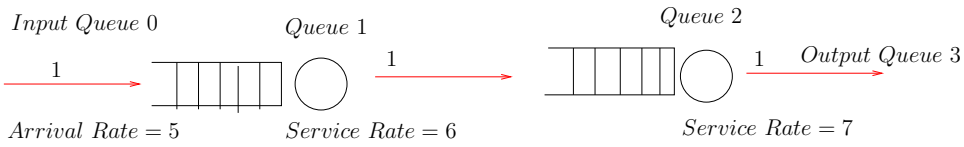

Figure 2: A Queueing Network with Two Queues

It can be shown that $W = 1.5$ for this system.

The above network does not have feedback i.e., a customer never revisits a queue more than once. Figure 3 gives an example of a network with feedback.

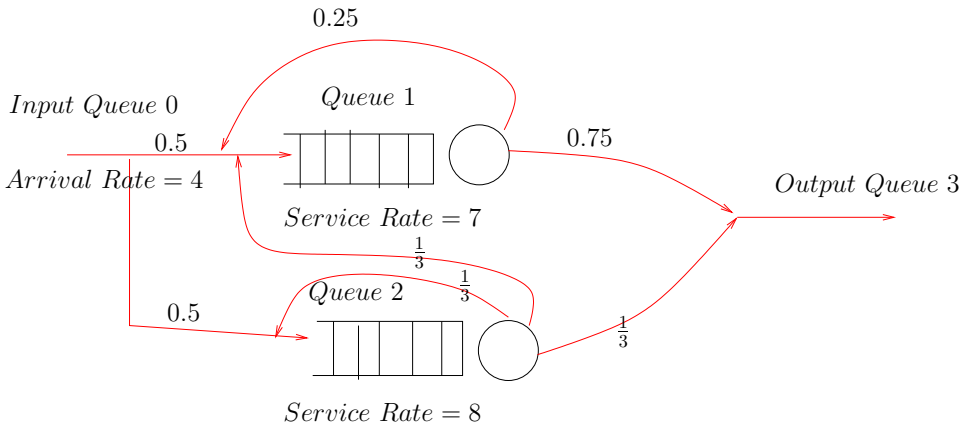

Figure 3: A Queueing Network with Feedback

In this network, the customers arrive at rate 5 at the input queue. Each arriving customer chooses whether to enter queue 1 or queue 2 with probability 0.5. When a customer is serviced on queue 1, it either goes to the output with probability 0.75, or makes a re-entry into queue 1 with probability 0.25. A customer which has completed service at queue 2 has three choices - (i) with probability 0.33 it re-enters queue 2, (ii) with probability 0.33 it enters queue 1, or (iii) with probability 0.33 it goes to the output.

The average time spent by a customer in this queueing network is $W = 0.48333$ seconds.

Clearly, we can construct more and more complex queueing networks if we allow the output of one queue to probabilistically go to the input of some other queue.

Let us define such a queueing system mathematically. There are total of m

queues $Q_0, Q_1, \cdots, Q_{m-1}$ in the system, out of which the first $n, n \geq 1$ queues are input queues. The last queue Q_{m-1} is an output queue, and there is always a unique output queue in a network. A customer enters the output queue only after it has been served by the network and hence leaves the system.

The following are the parameters of the system:

1. The input queues $Q_0, Q_1, \cdots, Q_{n-1}$ have inter-arrival rates $\lambda_0, \lambda_1, \cdots, \lambda_{n-1}$ respectively. This means that the time between the arrival of two consecutive customers at input queue i is exponentially distributed with mean $\frac{1}{\lambda_i}$.

2. The input-output queues $Q_n, Q_{n+1}, \cdots, Q_{m-2}$ have service rates $\lambda_n, \lambda_{n+1}, \cdots, \lambda_{m-2}$ respectively. This means that the service time of a job at the server of queue $Q_i, n \leq i \leq m-2$ is exponentially distributed with mean $\frac{1}{\lambda_i}$.

3. For each queue Q_i, except the output queue, we have $m-n$ probabilities $P_{in}, P_{i(n+1)}, \cdots, P_{im}$ such that $P_{in} + P_{i(n+1)} + \cdots + P_{im} = 1$. A job which has just finished at queue i, moves to queue j with probability P_{ij}.

4 Input Format

In this assignment, you will be given the parameters of a queueing network as described above. Your objective is to write a C++ program that simulates this system and calculates the average time W spent in the system by a customer.

The input format is as follows. The first input is a number t, which is the time for which you will run the simulation. We assume that the simulation starts at time 0, and all queues are empty at the the start. Thus, $t = 10000$ asks you to simulate the queueing network for 10000 time units.

The next two numbers m and n denote the total number of queues, and the number input queues, respectively. After this there are m numbers which correspond to the values of $\lambda_0, \lambda_1, \cdots, \lambda_{n-1}$ respectively.

The only data remaining are the transition probabilities. The input contains $n-1$ more lines, with each line encoding the transition probabilities for queue $i, 0 \leq i \leq n-2$ (The output queue has no transition probabilities).

The transition probability for queue i is given as follows. The first entry is a number k_i which is equal to the number of queues to which a job that finishes at job i can go. After this, we have k_i pairs of numbers of the form (j, P_{ij}), where the first number in a pair denotes the queue number j and the second number denotes the probability of entering queue j from queue i.

For example, the network in Figure 2 will be input as:

```
50000 // simulation time
4   // number of queues
1   // number of input queues
5 6 7 // lambda values for queues
1 1 1 // transition probabilities for queue 0
```

```
1 2 1 // transition probabilities for queue 1
1 3 1 // transition probabilities for queue 2
```

Similarly, the input for simulating the network in Figure 3 is:

```
100000
4
1
4 7 8
2 1 0.5 2 0.5
2 3 0.75 1 0.25
3 1 0.33333 2 0.33333 3 0.33333
```

Given an input in the above format, your program will output a single floating-point number W which is equal to the average time spent by a customer in the network during the simulation.

For example, your program when run on the above input should output:

```
0.48333
```

Note that since W is a random-variable, the value returned by your program may not be exactly 0.48333. However, it should be reasonably close to it (say within 10 percent). Two independent simulations of this queueing system gave $W = 0.4848$ and $W = 0.4850$ respectively.

As a final example, consider the queueing network of Figure 4. There are 5 queues, out of which 1 is an input queue, and 5 is the output queue. The queue numbers are shown in green. The inter-arrival rate at Q_0 is 1, and the service rates at queues Q_1, Q_2, Q_3, Q_4 are $3, 4, 5, 6$ respectively. The arrival and service rates are shown in blue.

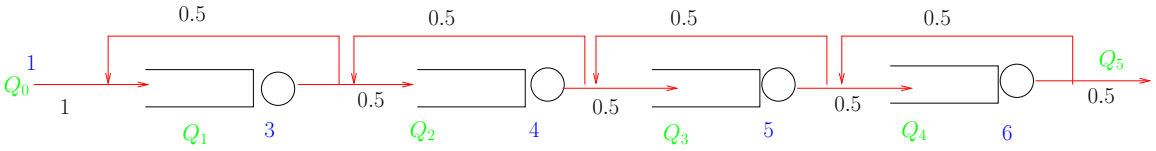

Figure 4: A more complex Queueing Network

All customers arriving at input queue Q_0 enter queue Q_1 with probability 1. A customer finishing at any other queue Q_i either (i) re-enters the same queue with probability 0.5, or (ii) goes to queue Q_{i+1} with probability 0.5.

Here is the input text for simulating this network for 50000 time units:

```
50000
6
1
1 3 4 5 6
```

```
1 1 1
2 1 0.5 2 0.5
2 2 0.5 3 0.5
2 3 0.5 4 0.5
2 4 0.5 5 0.5
```

The exact value of W for this network is 4.167. I ran two simulations of this network - one gave $W = 4.17782$, and the second gave $W = 4.22696$.