Queuing System Project Guidelines

Objective

Students will design and analyze a queuing system inspired by a real-life scenario. The goal is to extend the basic M/M/c queuing model by incorporating additional complexity, such as networks of queues or customer behaviors, and to evaluate system performance under different conditions.

Submission Requirements

Groups must submit:

- **Video Presentation** (max 6 minutes):
  - Explain the chosen scenario and its relevance.
  - Describe the queuing model (e.g., network structure, arrival rates, service rates).
  - Present the results of the analysis (e.g., metrics, comparisons).
  - Highlight key insights and recommendations.
  - Ensure the video includes human narration (AI-generated videos are not allowed).
- **Python Script**: Submit the Python script used to implement and analyze the queuing system.

Tasks

1. Identify a Relevant Scenario

Choose a real-life scenario for your queuing system, such as:

- **University Cafeteria**: Queues for ordering food and paying.
- **Public Transportation**: Ticket counters and boarding.
- **Healthcare**: Patients moving through different stages of a clinic visit.
- **Retail Store**: Queues for fitting rooms and checkout counters.

2. Extend the M/M/c Model

a. **Design a Network of Queues:**
   - Include multiple interconnected queues where customers transition between different stages.
   - Example: At a cafeteria, customers first queue to order and then queue to pay.

b. **Incorporate Customer Behaviors:**
   - Consider behaviors such as jockeying (switching lines) or reneging (leaving the queue).
   - Example: Customers leave if the wait time exceeds a threshold.

3. Compare Metrics Under Different Conditions

- Evaluate the system's performance using metrics such as:
  - Average queue length.
  - Average waiting time.
  - Server utilization.
- Analyze the impact of varying:
  - Arrival and service rates.
  - Customer behaviors.

  - Number of servers.

4. Develop and Test the System in Python Using SimPy

- Use the `SimPy` package in Python for simulation.
- Create a workflow to:
  - Define resources (e.g., servers, queues).
  - Simulate customer arrivals and service.
  - Collect metrics for analysis.
- Refer to examples and tutorials available at `https://simpy.readthedocs.io/en/latest/examples/index.html` for guidance and inspiration.

Example Scenario: Ambulatory Queuing System

Imagine a small ambulatory where patients go through two stages:

- **Stage 1: Nurse Assessment**
  - Patients first queue to be seen by a nurse.
  - Service time: Exponentially distributed with an average of 5 minutes.
  - Patients will not join this queue with a probability of 20% if there are more than 5 people waiting.
- **Stage 2: Doctor Consultation**
  - After seeing the nurse, patients proceed to queue for a doctor.
  - Service time: Exponentially distributed with an average of 10 minutes.

Arrival Process: Patients arrive at the ambulatory following a Poisson process with a mean rate of 12 patients per hour.

Suggested Analyses

Using SimPy, simulate the ambulatory system for an 8-hour workday (480 minutes) and perform the following analyses:

- **Queue Size Evolution:**
  - Plot the size of the nurse and doctor queues over time.
  - Highlight periods of congestion or idle times.
- **Reneging Statistics:**
  - Calculate the proportion of patients who leave the system without receiving service.
  - Identify times of peak reneging and possible causes.
- **Resource Utilization:**
  - Determine the utilization rates of the nurse and doctor.
  - Assess whether additional staff or resources are needed.
- **Waiting Times:**
  - Calculate the average waiting time for patients in the nurse and doctor queues.
  - Identify conditions under which waiting times exceed acceptable limits.