### Phần 1: Giới thiệu và Mục tiêu

**Giới thiệu**:  
Mô hình này mô phỏng hệ thống xếp hàng trong nhà hàng buffet, nơi khách hàng đến và lựa chọn các khu vực buffet khác nhau để phục vụ thức ăn. Mỗi khu vực có giới hạn về số lượng khách hàng có thể phục vụ cùng lúc.

**Mục tiêu**:  
Xây dựng mô hình mô phỏng để đánh giá các thông số:

* Thời gian chờ trung bình của khách tại mỗi khu vực.
* Thời gian phục vụ trung bình.
* Khả năng tối ưu hóa hiệu năng thông qua phân phối khách hàng giữa các khu vực.

### Phần 2: Thiết lập và Thiết kế Hệ thống Xếp Hàng

**Tham số mô hình**

* + arrival\_rate (Tỷ lệ đến): Tỷ lệ khách hàng đến nhà hàng.
  + service\_rate (Tỷ lệ phục vụ): Tỷ lệ phục vụ khách tại các khu vực buffet.
  + num\_stations (Số lượng khu vực phục vụ): Tổng số khu vực buffet trong nhà hàng.
  + station\_capacity (Dung lượng khu vực): Số lượng khách hàng tối đa có thể chờ trong hàng đợi tại mỗi khu vực.

**Thiết kế mô hình xếp hàng**

* + Mỗi khu vực phục vụ sẽ được mô hình hóa như một hàng đợi M/M/c/K với:
    - c là số máy chủ tại mỗi khu vực.
    - K là số lượng tối đa khách hàng có thể xếp hàng tại mỗi khu vực.
  + Các khách hàng đến sẽ được phân phối ngẫu nhiên tới các khu vực để nhận phục vụ.

### Phần 3: Xây dựng và Triển khai Mô phỏng bằng SimPY

#### 3.1. Cài đặt thư viện

Trước tiên, hãy đảm bảo rằng SimPY đã được cài đặt:

python

Copy code

pip install simpy

#### 3.2. Mô phỏng hệ thống buffet

python

Copy code

import simpyimport randomimport pandas as pd

# Thiết lập tham số

arrival\_rate = 1/2 # Trung bình mỗi 2 phút có 1 khách đến

service\_rate = 1/3 # Trung bình mất 3 phút để phục vụ mỗi khách

num\_stations = 3 # Số khu vực buffet

station\_capacity = 5 # Số khách tối đa trong hàng đợi mỗi khu vực

# Lưu trữ dữ liệu

wait\_times = []

service\_times = []

# Hàm mô phỏng quy trình phục vụ tại một khu vựcdef serve\_customer(env, service\_time):

start\_time = env.now

yield env.timeout(service\_time) # Đợi để hoàn thành phục vụ

end\_time = env.now

service\_times.append(end\_time - start\_time)

# Hàm mô phỏng khách hàng đến nhà hàngdef customer\_arrivals(env, stations):

while True:

interarrival\_time = random.expovariate(1.0 / arrival\_rate)

yield env.timeout(interarrival\_time) # Khách đến sau khoảng thời gian ngẫu nhiên

# Chọn ngẫu nhiên một khu vực buffet

selected\_station = random.choice(stations)

# Nếu hàng đợi chưa đầy, khách sẽ vào hàng đợi

if len(selected\_station.queue) < station\_capacity:

arrival\_time = env.now

with selected\_station.request() as req:

yield req # Chờ đến khi đến lượt phục vụ

wait\_times.append(env.now - arrival\_time)

env.process(serve\_customer(env, 1/service\_rate))

# Khởi tạo môi trường mô phỏng và các khu vực buffet

env = simpy.Environment()

stations = [simpy.Resource(env, capacity=1) for \_ in range(num\_stations)]

# Bắt đầu mô phỏng

env.process(customer\_arrivals(env, stations))

env.run(until=120) # Chạy mô phỏng trong 120 đơn vị thời gian

# Phân tích dữ liệu

data = pd.DataFrame({

'Wait Times': wait\_times,

'Service Times': service\_times

})

print(data.describe()) # Thống kê dữ liệu

### Phần 4: Phân tích và So sánh Kết quả với Công thức Toán học

Sau khi chạy mô phỏng, chúng ta sẽ phân tích dữ liệu thu thập được để so sánh với các công thức toán học cơ bản. Một số công thức quan trọng bao gồm:

* **Luật Little**: L=λ⋅WL = \lambda \cdot WL=λ⋅W, với LLL là số khách trung bình trong hệ thống, λ\lambdaλ là tỷ lệ đến, và WWW là thời gian chờ trung bình.

#### Phân tích dữ liệu

Sử dụng dữ liệu từ data.describe() để:

1. Tính thời gian chờ trung bình.
2. Tính thời gian phục vụ trung bình.
3. So sánh với các giá trị lý thuyết từ công thức toán học.

#### Kiểm tra Kết quả

So sánh các kết quả mô phỏng với công thức toán học để xác minh độ chính xác của mô hình. Nếu thời gian chờ và thời gian phục vụ trung bình phù hợp, mô hình được xem là hợp lệ.

### Phần 5: Báo cáo

**Giới thiệu**

* + Mô tả bài toán hàng đợi trong nhà hàng buffet và mục tiêu của dự án.

**Thiết kế hệ thống**

* + Mô tả cấu trúc và tham số của hệ thống buffet.
  + Trình bày lý do lựa chọn mô hình M/M/c/K cho các khu vực buffet.

**Kết quả mô phỏng**

* + Trình bày kết quả mô phỏng, bao gồm thời gian chờ và thời gian phục vụ trung bình.
  + Trình bày các bảng số liệu và biểu đồ (nếu cần).

**Phân tích và đánh giá**

* + So sánh các kết quả thực nghiệm với lý thuyết (luật Little).
  + Nhận xét về hiệu quả của từng khu vực buffet.

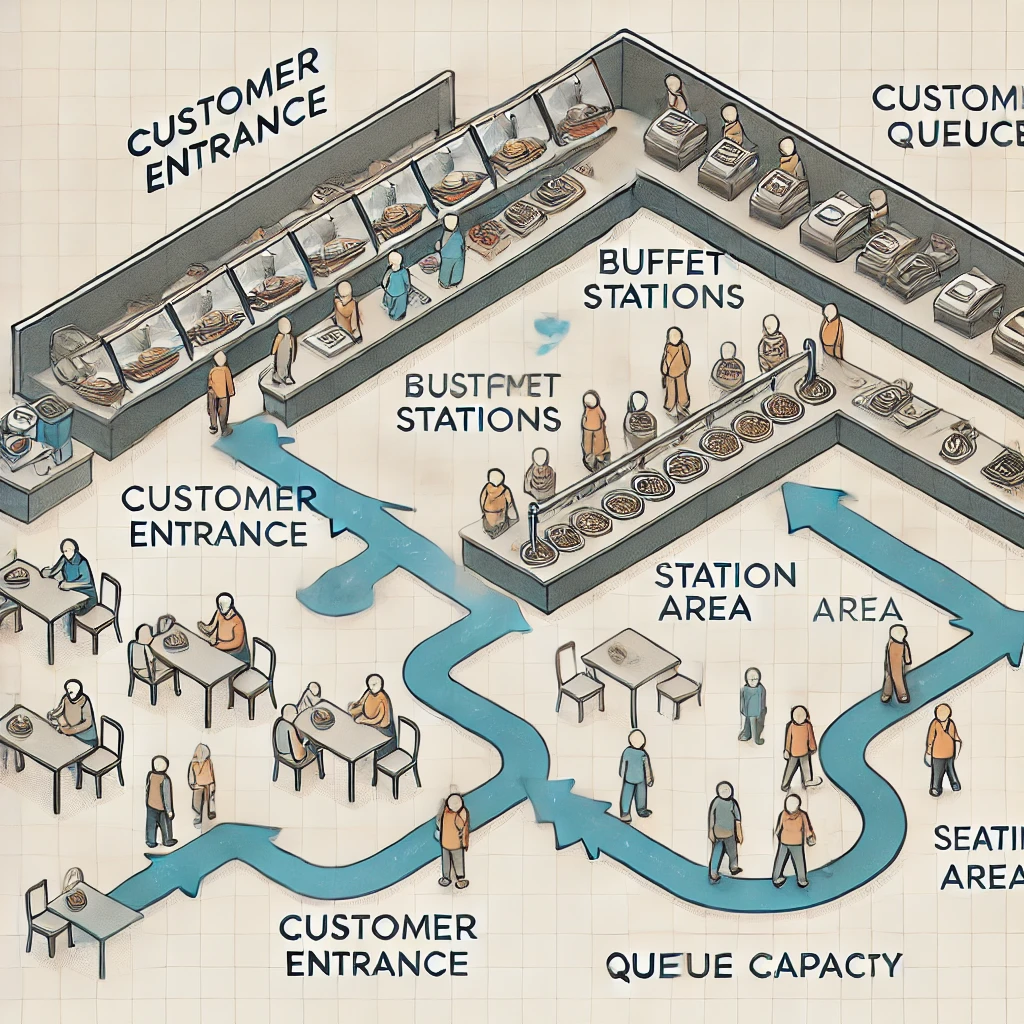
**Kết luận**

* + Tóm tắt những phát hiện chính và đề xuất các biện pháp cải thiện hiệu quả của hệ thống, chẳng hạn như tăng số lượng máy chủ ở những khu vực thường xuyên có thời gian chờ cao.

Với các phần trên, bạn có thể hoàn thành một báo cáo chi tiết và thực hiện mô phỏng đầy đủ. Nếu cần hỗ trợ thêm trong việc giải thích hoặc điều chỉnh mô hình, hãy cho tôi biết để tôi có thể giúp đỡ thêm.

Để hoàn thành phần **Outcomes** của hệ thống mô phỏng hàng đợi trong nhà hàng buffet, chúng ta sẽ tạo bản vẽ thiết kế sơ đồ hệ thống và mô tả chi tiết các hàm cần triển khai, bao gồm thông tin về **input**, **output**, và **chức năng** của từng hàm.

### Bản vẽ Thiết kế Hệ thống (Sơ đồ Kiến trúc)



**Sơ đồ kiến trúc** của hệ thống mô phỏng nhà hàng buffet sẽ bao gồm các thành phần chính sau:

* **Customer Generator**: Phát sinh khách hàng mới vào hệ thống.
* **Queue (Hàng đợi)**: Mô phỏng hàng đợi tại mỗi khu vực buffet. Mỗi khu vực có giới hạn số lượng khách hàng trong hàng đợi.
* **Service Stations (Khu vực phục vụ)**: Các khu vực phục vụ đồ ăn tại buffet. Mỗi khu vực có một hoặc nhiều máy chủ để phục vụ khách hàng.
* **Event Logger**: Ghi nhận và lưu trữ các sự kiện như thời gian chờ và thời gian phục vụ của từng khách hàng.

**Sơ đồ luồng dữ liệu**:

rust

Copy code

Customer Generator --> Queue (Hàng đợi) --> Service Stations --> Event Logger

### 2. Mô tả Các Hàm Cần Triển Khai

#### Hàm 1: serve\_customer

* **Chức năng**: Thực hiện quá trình phục vụ khách hàng tại một khu vực buffet.
* **Input**:
  + env (simpy.Environment): Môi trường mô phỏng.
  + service\_time (float): Thời gian phục vụ cho một khách hàng.
* **Output**: Không có giá trị trả về. Thời gian phục vụ của khách sẽ được ghi nhận vào danh sách service\_times.
* **Mô tả**:
  + Hàm này mô phỏng quá trình phục vụ của một khách hàng tại một khu vực buffet.
  + Hàm sử dụng yield env.timeout(service\_time) để mô phỏng thời gian phục vụ khách.
  + Ghi nhận thời gian bắt đầu và kết thúc phục vụ để tính toán thời gian phục vụ thực tế.

#### Hàm 2: customer\_arrivals

* **Chức năng**: Phát sinh các khách hàng đến nhà hàng buffet và phân phối họ vào các khu vực phục vụ.
* **Input**:
  + env (simpy.Environment): Môi trường mô phỏng.
  + stations (list): Danh sách các khu vực phục vụ (tương ứng với simpy.Resource cho từng khu vực buffet).
* **Output**: Không có giá trị trả về, nhưng thời gian chờ của mỗi khách sẽ được lưu vào danh sách wait\_times.
* **Mô tả**:
  + Hàm này tạo khách hàng mới với khoảng thời gian ngẫu nhiên dựa trên arrival\_rate.
  + Mỗi khách hàng được gán vào một khu vực phục vụ ngẫu nhiên.
  + Nếu hàng đợi tại khu vực đó chưa đầy, khách sẽ được đưa vào hàng đợi và đợi đến lượt phục vụ.
  + Thời gian chờ của khách hàng được tính bằng env.now - arrival\_time.

#### Hàm 3: init\_simulation

* **Chức năng**: Thiết lập và khởi động môi trường mô phỏng với các tham số đầu vào.
* **Input**:
  + arrival\_rate (float): Tỷ lệ đến của khách hàng.
  + service\_rate (float): Tỷ lệ phục vụ của mỗi máy chủ.
  + num\_stations (int): Số lượng khu vực phục vụ.
  + station\_capacity (int): Dung lượng hàng đợi tối đa của mỗi khu vực.
  + simulation\_time (float): Thời gian chạy mô phỏng.
* **Output**: Không có giá trị trả về, chỉ khởi động môi trường mô phỏng.
* **Mô tả**:
  + Khởi tạo môi trường simpy.Environment và danh sách các khu vực phục vụ.
  + Tạo và bắt đầu quá trình mô phỏng khách hàng đến bằng cách gọi customer\_arrivals.
  + Chạy mô phỏng cho đến thời gian simulation\_time.

#### Hàm 4: log\_event\_data

* **Chức năng**: Lưu trữ dữ liệu sự kiện như thời gian chờ và thời gian phục vụ.
* **Input**:
  + wait\_times (list): Danh sách thời gian chờ của khách hàng.
  + service\_times (list): Danh sách thời gian phục vụ của khách hàng.
* **Output**: Không có giá trị trả về. Hàm ghi lại dữ liệu vào file hoặc in ra console.
* **Mô tả**:
  + Hàm này có thể lưu trữ dữ liệu vào file hoặc hiển thị ra màn hình.
  + Tạo báo cáo tóm tắt từ các dữ liệu đã thu thập.

#### Hàm 5: analyze\_results

* **Chức năng**: Phân tích dữ liệu thu thập từ mô phỏng và tính toán các thống kê như thời gian chờ trung bình, thời gian phục vụ trung bình.
* **Input**:
  + data (pandas.DataFrame): Bảng dữ liệu chứa thời gian chờ và thời gian phục vụ.
* **Output**: Kết quả phân tích thống kê như trung bình, phương sai, v.v.
* **Mô tả**:
  + Hàm này sử dụng pandas để tính toán các giá trị thống kê như thời gian chờ trung bình và thời gian phục vụ trung bình.
  + Kết quả sẽ được so sánh với các công thức toán học như luật Little để xác nhận tính chính xác của mô phỏng.

### 3. Ví dụ Mã Mô phỏng

Dưới đây là ví dụ mã sử dụng các hàm trên để thực hiện mô phỏng:

python

Copy code

import simpyimport randomimport pandas as pd

wait\_times = []

service\_times = []

# Hàm phục vụdef serve\_customer(env, service\_time):

start\_time = env.now

yield env.timeout(service\_time)

service\_times.append(env.now - start\_time)

# Hàm phát sinh khách hàngdef customer\_arrivals(env, stations):

while True:

interarrival\_time = random.expovariate(1.0 / arrival\_rate)

yield env.timeout(interarrival\_time)

selected\_station = random.choice(stations)

if len(selected\_station.queue) < station\_capacity:

arrival\_time = env.now

with selected\_station.request() as req:

yield req

wait\_times.append(env.now - arrival\_time)

env.process(serve\_customer(env, 1/service\_rate))

# Hàm khởi tạo mô phỏngdef init\_simulation(arrival\_rate, service\_rate, num\_stations, station\_capacity, simulation\_time):

env = simpy.Environment()

stations = [simpy.Resource(env, capacity=1) for \_ in range(num\_stations)]

env.process(customer\_arrivals(env, stations))

env.run(until=simulation\_time)

# Hàm lưu dữ liệudef log\_event\_data():

data = pd.DataFrame({'Wait Times': wait\_times, 'Service Times': service\_times})

print(data.describe())

# Hàm phân tích dữ liệudef analyze\_results():

data = pd.DataFrame({'Wait Times': wait\_times, 'Service Times': service\_times})

print("Average wait time:", data['Wait Times'].mean())

print("Average service time:", data['Service Times'].mean())

# Thực hiện mô phỏng

arrival\_rate = 1/2

service\_rate = 1/3

num\_stations = 3

station\_capacity = 5

simulation\_time = 120

init\_simulation(arrival\_rate, service\_rate, num\_stations, station\_capacity, simulation\_time)

log\_event\_data()

analyze\_results()

### Tóm tắt

* **Bản vẽ thiết kế**: Phác thảo sơ đồ kiến trúc của hệ thống.
* **Các hàm mô phỏng**: Đã xác định và triển khai các hàm phục vụ, phát sinh khách hàng, khởi tạo mô phỏng, lưu dữ liệu, và phân tích kết quả.

Các phần này sẽ giúp hoàn thiện phần Outcomes theo yêu cầu của đề bài.

Dưới đây là bản thiết kế sơ đồ luồng (flowchart) cho mô hình hàng đợi trong nhà hàng buffet, dựa trên yêu cầu đề bài:

**Bắt đầu (Start)**: Khách hàng bắt đầu tiến vào hệ thống hàng đợi của nhà hàng buffet.

**Kiểm tra Khu vực Phục vụ (Check Buffet Stations)**:

* 1. Khách hàng sẽ chọn một trong các khu vực phục vụ đồ ăn.
  2. Nếu hàng đợi của khu vực đầy, khách hàng sẽ phải đợi hoặc chuyển sang khu vực khác (phụ thuộc vào mô hình mô phỏng).

**Xếp Hàng Chờ (Join Queue)**:

* 1. Khách hàng sẽ xếp hàng trong hàng đợi của khu vực đã chọn.

**Phục vụ Khách (Serve Customer)**:

* 1. Khi đến lượt, khách hàng sẽ được phục vụ.
  2. Thời gian phục vụ được xác định dựa trên service\_rate và station\_capacity.

**Ghi Lại Dữ Liệu Sự Kiện (Log Event Data)**:

* 1. Thời gian chờ và thời gian phục vụ của khách hàng sẽ được ghi nhận để phân tích sau này.

**Kết Thúc (End)**: Sau khi được phục vụ, khách hàng rời khỏi hàng đợi và di chuyển đến khu vực ngồi hoặc ra khỏi hệ thống.

