2. Queue Design — การออกแบบโครงสร้างคิว

ระบบ Food Delivery Order Queue ออกแบบโครงสร้างคิวไว้ 2 **รูปแบบ** เพื่อใช้เปรียบเทียบประสิทธิภาพในการจัดลำดับคำสั่งซื้อ ได้แก่ FIFO Queue และ Priority Queue

2.1 Algorithm A: FIFO Queue

ใช้โครงสร้างข้อมูล Queue แบบ FIFO (First In, First Out) โดย Order ที่เข้ามาก่อนจะถูกประมวลผลก่อน และ Order ที่เข้ามาทีหลังจะรออยู่ด้านหลังคิว

ใน Java ใช้ `ArrayDeque<Order>` เป็นโครงสร้างสำหรับเก็บ Order

ตัวอย่างลำดับการเข้าคิว:

O1 → O2 → O3 → O4 → O5

ดังนั้นเมื่อทำการ DEQUEUE จะนำ Order ออกจากด้านหน้าคิวตามลำดับ

ข้อดี: มีความยุติธรรม เพราะทุก Order ได้รับการประมวลผลตามลำดับที่เข้ามา และการเพิ่ม/นำ Order ออกจากคิวทำได้รวดเร็ว

2.2 Algorithm B: Priority Queue

ใช้โครงสร้างข้อมูล Priority Queue โดยไม่ได้ประมวลผลตามลำดับการเข้ามาเพียงอย่างเดียว แต่จะเลือก Order ที่มี Priority **สูงกว่า** ก่อน

ใน Java ใช้ `PriorityQueue<Order>` และกำหนดเงื่อนไขการจัดลำดับดังนี้

1. พิจารณา Priority ของ Order
2. หาก Priority เท่ากัน ให้พิจารณา Order Time
3. Order ที่มี Priority สูงกว่าจะถูกนำไปประมวลผลก่อน

ตัวอย่างลำดับการประมวลผล:

O2 → O4 → O1 → O3 → O5

โดย O2 และ O4 เป็น Express จึงได้รับการประมวลผลก่อน Normal และ Large

ข้อดี: ช่วยลดเวลารอของ Order ที่มีความสำคัญสูง เช่น Express

ข้อจำกัด: หากมี Express เข้ามาอย่างต่อเนื่อง Order ที่ Priority ต่ำกว่าอาจต้องรอนาน หรือเกิด Starvation ได้

2.3 ข้อมูลที่จัดเก็บใน Order

Field	รายละเอียด
Order ID	รหัสคำสั่งซื้อ
Order Time	เวลาที่ Order เข้ามา
Food Type	ประเภทอาหาร
Preparation Time	เวลาเตรียมอาหาร
Priority	ระดับความสำคัญ

สรุป: Algorithm A เน้นความยุติธรรม โดยประมวลผล Order ตามลำดับที่เข้ามา ส่วน Algorithm B เน้นการให้ความสำคัญกับ Order ที่มี Priority สูงก่อน เพื่อให้สามารถเปรียบเทียบรูปแบบการจัดคิวของทั้งสอง Algorithm ได้อย่างเหมาะสมสำหรับระบบ Food Delivery