

CHAPTER 5 QUEUE

ผศ.ดร.สิดดา อินทรโสธรฉันท

สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น

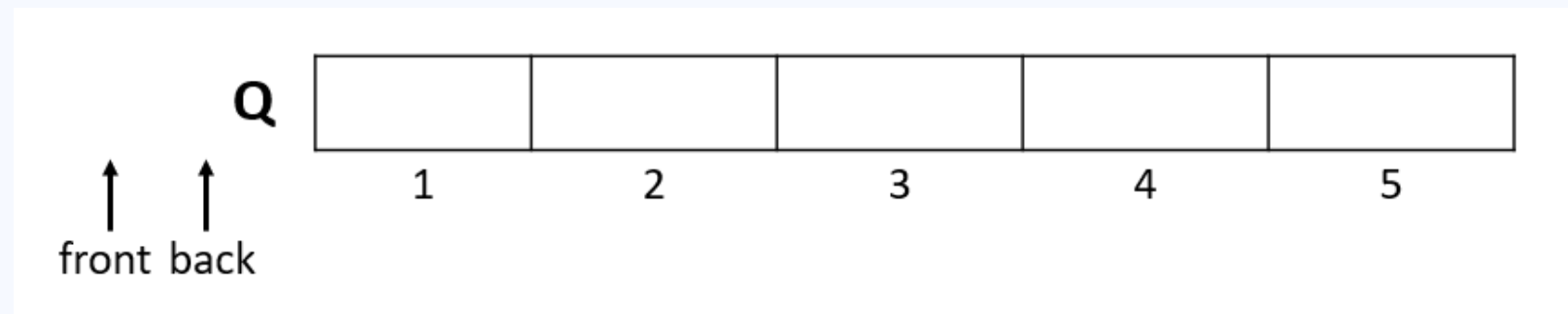
คุณสมบัติของ QUEUE

- การจัดการข้อมูลเข้าออกจะเป็นรูปแบบที่ข้อมูลที่เข้าก่อนจะออกก่อน ตามลำดับ
- การทำงานของคิวจะมีลักษณะคล้ายท่อที่ทางเข้าและออกอยู่คนละทาง
- การจัดการคิวจะมีตัวแปรที่ช่วยควบคุมการทำงาน 2 ตัว คือ ตัวแปรที่ช่วยจัดการในการนำข้อมูลเข้าเพื่อต่อท้ายคิวและตัวแปรที่ช่วยจัดการในการนำข้อมูลออกที่ตำแหน่งหัวของคิว

คุณสมบัติของ QUEUE

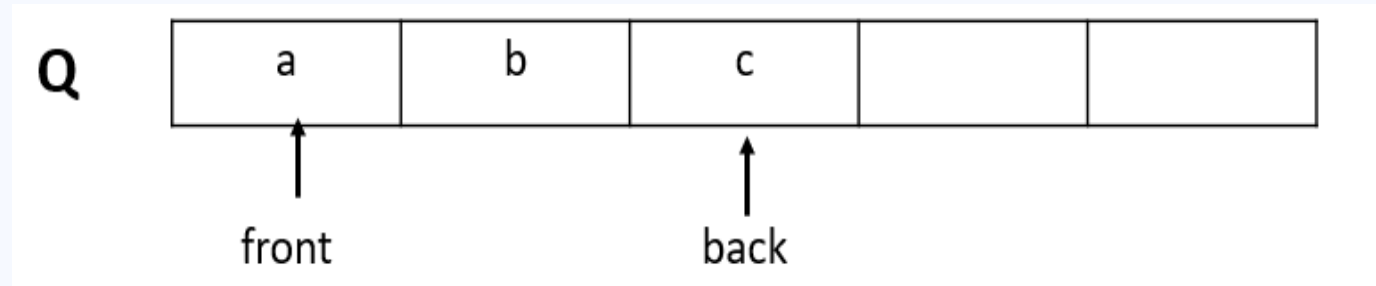
- จากภาพ ลักษณะของคิวจะมีลักษณะเป็นท่อที่สามารถรับข้อมูลได้จำนวน 5 ตำแหน่ง
- ตัวแปรช่วยในการควบคุมการทำงานของคิวอยู่ 2 ตัวคือ
 - ตัวแปร front จะทำหน้าที่ชี้ตำแหน่งแรกของข้อมูล
 - ตัวแปร back จะทำหน้าที่ชี้ตำแหน่งสุดท้ายของข้อมูล
- หากไม่มีข้อมูลในคิว ตัวแปรทั้ง 2 จะมีค่าเป็น 0

คุณสมบัติของ QUEUE



- เมื่อมีการนำข้อมูลออกข้อมูลที่ตำแหน่ง front จะถูกนำออกไป
- ในขณะที่การนำข้อมูลเข้าจะเป็นการนำข้อมูลมาต่อท้ายตำแหน่ง back

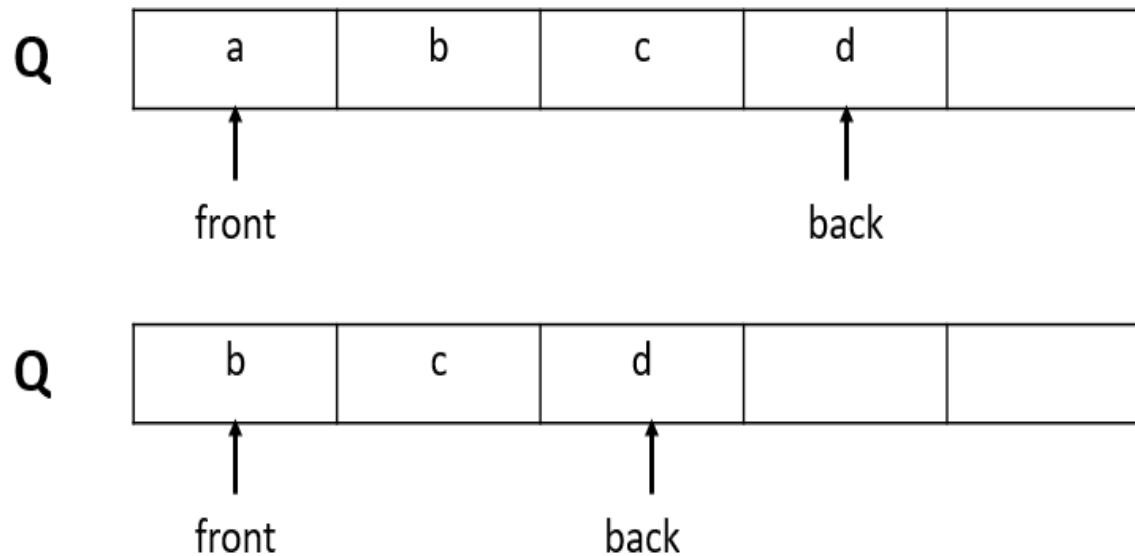
คุณสมบัติของ QUEUE



- ข้อมูลใหม่จะถูกนำมาใส่ต่อท้าย c นั่นเอง
- จากภาพ “a” จะถูกนำออกก่อน แล้วตามด้วย b และ c ตามลำดับ เมื่อมีการนำข้อมูลออกทุกครั้ง ข้อมูลที่เหลือในคิวจะถูกเลื่อนตำแหน่งเข้าหาตำแหน่งแรกของคิว จึงเรียกคิวประเภทนี้ว่า **คิวแบบตรงหัวคิว**

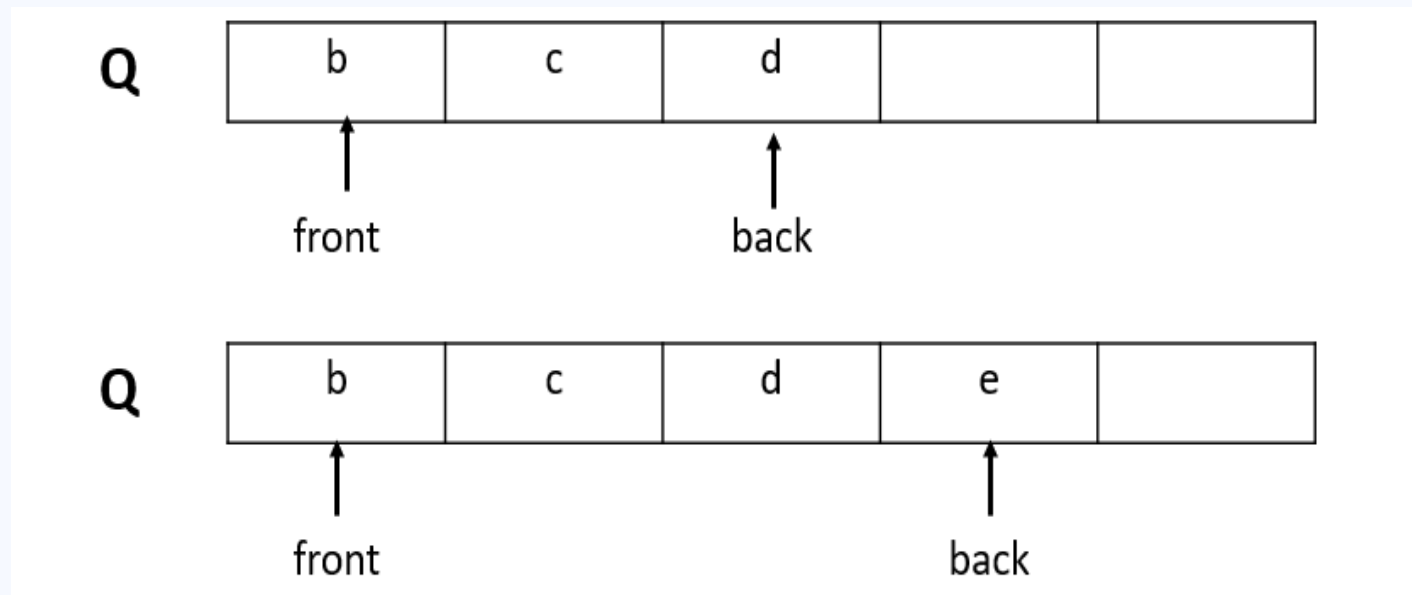
คิวแบบตรึงหัวคิว

- เมื่อมีข้อมูลออก ข้อมูลที่เหลือในคิวจะถูกเลื่อนตำแหน่งเข้าหาตำแหน่งแรกของคิว เพื่อแทนที่ข้อมูลเดิมที่ถูกนำออกไป นั่นคือ ตำแหน่งแรกของคิวถูกตรึงไว้นั่นเอง



คิวแบบตรึงหัวคิว

- การนำข้อมูลเข้านั้นจะเป็นการทำงานที่ตำแหน่งท้ายของคิว โดยข้อมูลใหม่ที่น่าเข้ามาจะอยู่ตำแหน่งต่อท้ายหลังตำแหน่ง back



ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิวแบบตรึงหัวคิว

- กำหนด คิว Q สามารถจุข้อมูลได้ N ตัว
- ตัวแปร front ทำหน้าที่ชี้ตำแหน่งหัวคิว
- ตัวแปร back ทำหน้าที่ชี้ตำแหน่งท้ายคิว

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิวแบบตรึงหัวคิว

enqueue (item)

1. ตรวจสอบว่าคิวเต็มหรือไม่
2. ถ้าคิวยังไม่เต็ม จะดำเนินการดังนี้
 - 2.1 คิวมีข้อมูลเดิมอยู่หรือไม่ ถ้าไม่มีข้อมูลเดิมอยู่ ให้กำหนดค่าเริ่มต้นให้ front เป็นตำแหน่งแรก

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิวแบบตเรียงหัวคิว

2.2 ขยับ back ไปตำแหน่งถัดไป : $back = back + 1$

2.3 ใส่ข้อมูล item ลงตำแหน่ง back : $Q[back] = item$

3. ถ้าคิวเต็ม ก็ไม่สามารถใส่ข้อมูลลงไปได้

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิวแบบตรึงหัวคิว

dequeue()

1. ตรวจสอบว่ามีข้อมูลในคิวหรือไม่ ซึ่งสามารถตรวจสอบได้จากตัวแปร front และ back
2. ถ้ามีข้อมูลในคิวจะทำการคัดลอกข้อมูลที่ตำแหน่ง front ไว้

item = Q[front]

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิวแบบตรึงหัวคิว

3. ทำการขยับข้อมูลที่เหลื้อมาแทนที่ข้อมูลที่ออกไป โดยมีขั้นตอนดังนี้

3.1 Repeat for $K = \text{front} + 1$ to back

$$Q[K-1] = Q[K]$$

3.2 ปรับค่า back ลดลง 1 ตำแหน่ง : $\text{back} = \text{back} - 1$

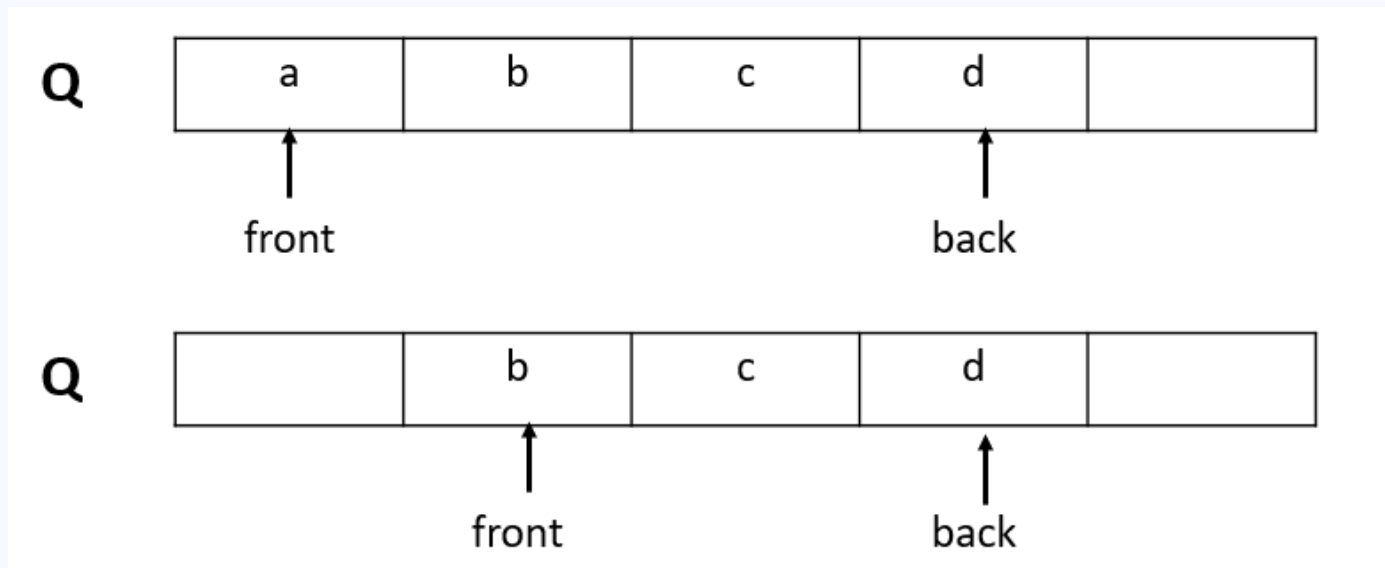
4. ส่งค่าที่คัดลอกไว้กลับไป : return item

คิวแบบไม่เรียงหัวคิว

- จากหลักการทำงานของคิวแบบตรึงหัวคิวจะพบว่า หากข้อมูลในคิวมีจำนวนมากหรือมีการนำข้อมูลออกบ่อย ทำให้ต้องเสียเวลามากในการปรับตำแหน่งข้อมูลในทุก ๆ ครั้งที่มีการทำงาน
- จึงมีการปรับการทำงานในส่วนของการนำข้อมูลออกให้เร็วขึ้น โดยแทนที่ทุกครั้งที่มีการนำข้อมูลออก จะต้องทำการปรับตำแหน่งข้อมูลที่เหลือ แต่ใช้วิธีปรับค่าตัวแปร front ไปหาสมาชิกตัวถัดไปแทน

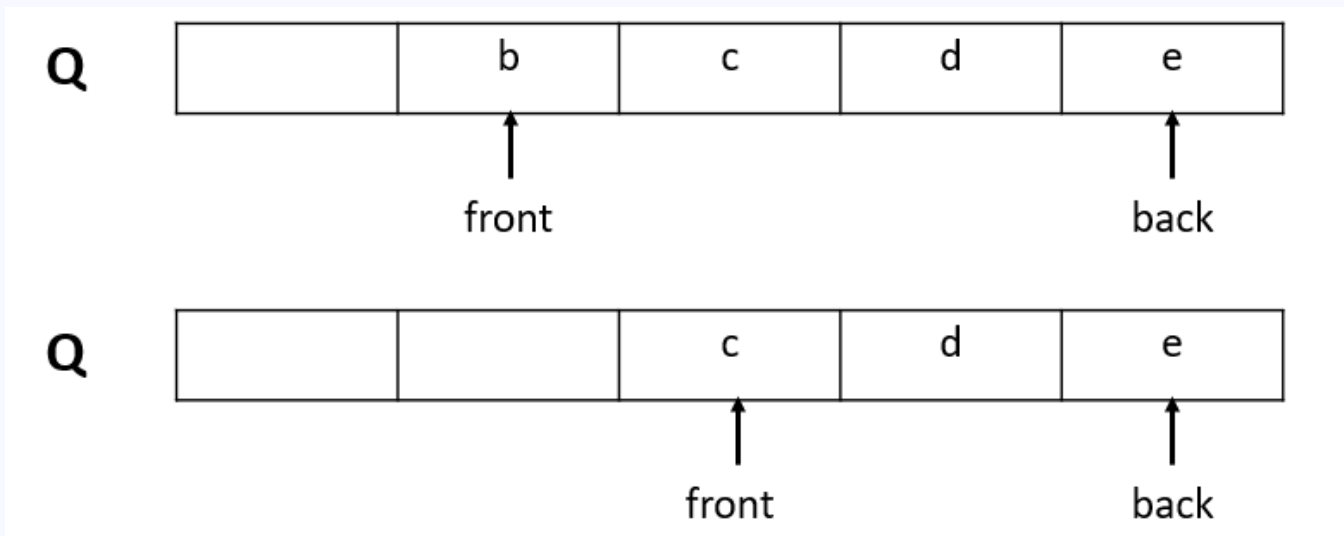
คิวแบบไม่เรียงหัวคิว

- เมื่อมีการนำข้อมูลออก “a” จะถูกนำออกก่อน หลังจากนั้นตัวแปร front จะเลื่อนตำแหน่งไปหาข้อมูลตัวถัดไปแทน



คิวแบบไม่เรียงหัวคิว

- การนำข้อมูลเข้ายังเหมือนกับคิวแบบตั้งหัวคิว คือ ต่อท้ายหลังตำแหน่งตัวแปร “back”



คิวแบบไม่ตรงหัวคิว

- จะเห็นว่าวิธีการนี้ช่วยให้การนำข้อมูลออกทำได้ง่ายขึ้น แต่ทุกครั้งที่มีการนำข้อมูลออกพื้นที่ที่เคยเก็บข้อมูลจะเสียทิ้งไป
- ดังนั้นการนำคิวแบบไม่ตรงไปใช้จึงเหมาะกับงานที่เน้นความรวดเร็ว และเก็บข้อมูลที่มีจำนวนค่อนข้างแน่นชัด เพื่อให้สามารถจองเนื้อที่ได้เพียงพอต่อการใช้งาน

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิวแบบไม่เรียงหัวคิว

- กำหนด คิว Q สามารถจุข้อมูลได้ N ตัว
- ตัวแปร front ทำหน้าที่ชี้ตำแหน่งหัวคิว
- ตัวแปร back ทำหน้าที่ชี้ตำแหน่งท้ายคิว

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิวแบบไม่เรียงหัวคิว

enqueue (item)

1. ตรวจสอบว่าคิวเต็มหรือไม่

2. ถ้าคิวยังไม่เต็ม จะดำเนินการดังนี้

- 2.1 คิวมีข้อมูลเดิมอยู่หรือไม่ ถ้าไม่มีข้อมูลเดิมอยู่ ให้กำหนดค่าเริ่มต้นให้ front เป็นตำแหน่งแรก

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิวแบบไม่เรียงหัวคิว

2.2 ขยับ back ไปตำแหน่งถัดไป : $back = back + 1$

2.3 ใส่ข้อมูล item ลงตำแหน่ง back : $Q[back] = item$

3. ถ้าคิวเต็ม ก็ไม่สามารถใส่ข้อมูลลงไปได้

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิวแบบไม่เรียงหัวคิว

dequeue()

1. ตรวจสอบว่ามีข้อมูลในคิวหรือไม่ ซึ่งสามารถตรวจสอบได้จากตัวแปร front และ back
2. ถ้ามีข้อมูลในคิวจะทำการคัดลอกข้อมูลที่ตำแหน่ง front ไว้

$item = Q[front]$

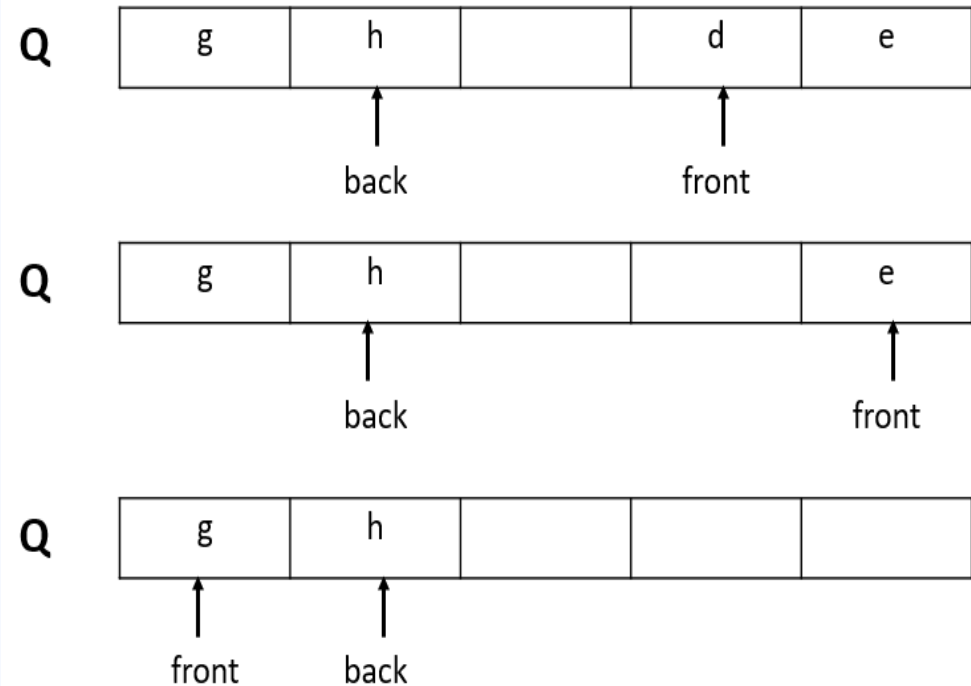
3. ปรับค่า front เลื่อนออกไป 1 ตำแหน่ง : $front = front + 1$
4. ส่งค่าที่คัดลอกไว้กลับไป : $return\ item$

คิวงกลม

- จากการทำงานของคิวแบบไม่เรียงหัวคิวที่ช่วยให้การนำข้อมูลออกทำได้รวดเร็วขึ้น แต่ยังมีข้อเสียในเรื่องของการสูญเสียพื้นที่ในการจัดเก็บข้อมูลในทุกครั้งที่มีการนำข้อมูลออก
- เกิดแนวคิดในการนำพื้นที่ส่วนต้นของคิวที่ว่างไว้มากำไว้ให้เกิดประโยชน์
- โดยนำกลับมาใช้ใหม่ เมื่อมีการทำงานถึงตำแหน่งสุดท้ายแล้ว จึงเรียกเรียกคิวประเภทนี้ว่า คิวงกลม

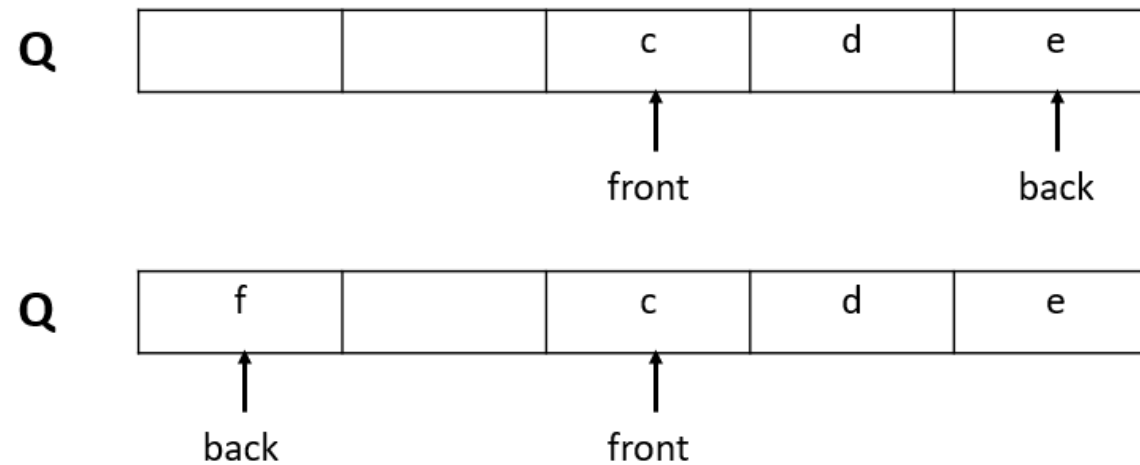
คิวงกลม

- การนำข้อมูลออกจากคิวงกลมจะมีการทำงานคล้ายคิวแบบไม่เรียงหัวคิว นั่นคือ เมื่อมีการนำข้อมูลที่ตำแหน่ง front ออก ตัวแปร front จะเลื่อนไปที่ตำแหน่งของข้อมูลตัวถัดไป แต่จะแตกต่างกันในกรณีที่เมื่อนำข้อมูลออกจนมาถึงตำแหน่งสุดท้ายแล้วก็จะวนมาที่ตำแหน่งแรกต่อไป



คิววงกลม

- การนำข้อมูลเข้าก็จะมีหลักการทำงานคล้ายการนำข้อมูลออก นั่นคือ เมื่อตัวแปร back อยู่ตำแหน่งสุดท้ายแล้วมีการนำข้อมูลเข้ามาใหม่ ตำแหน่งต่อไปที่จะใส่ข้อมูลก็จะกลับมาพิจารณาตำแหน่งแรกว่าว่างหรือไม่ หากยังว่างก็จะนำข้อมูลใส่ลงตำแหน่งดังกล่าว



ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิววงกลม

- กำหนด คิว Q สามารถจุข้อมูลได้ N ตัว
- ตัวแปร front ทำหน้าที่ชี้ตำแหน่งหัวคิว
- ตัวแปร back ทำหน้าที่ชี้ตำแหน่งท้ายคิว

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิววงกลม

enqueue (item)

1. ตรวจสอบว่าคิวเต็มหรือไม่ โดยให้ตรวจสอบว่า เมื่อ back ขยับไปตำแหน่งถัดไปแล้ว ตำแหน่งดังกล่าวเป็นตำแหน่ง front หรือไม่

1.1 ถ้าเป็นตำแหน่ง front แปลว่า คิวเต็มแล้ว

1.2 ถ้าไม่ใช่ตำแหน่ง front แปลว่า ยังมีที่ว่างอยู่ / ยังไม่เต็ม

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิววงกลม

2. ถ้าคิวยังไม่เต็ม จะขยับ back ไปตำแหน่งถัดไป โดยในการปรับตำแหน่งจะต้องตรวจสอบว่า back อยู่ตำแหน่งสุดท้ายแล้วหรือยัง

2.1 ถ้าอยู่ตำแหน่งสุดท้าย back จะกลายเป็นตำแหน่งแรก

2.2 ถ้าไม่ได้อยู่ตำแหน่งสุดท้าย $back = back + 1$

3. ใส่ข้อมูล item ลงตำแหน่ง back : $Q[back] = item$

4. ถ้าคิวเต็ม ก็ไม่สามารถใส่ข้อมูลลงไปได้

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิววงกลม

dequeue()

1. ตรวจสอบว่ามีข้อมูลในคิวหรือไม่ ซึ่งสามารถตรวจสอบได้จากตัวแปร front และ back ถ้า front หรือ back มีค่าเป็น 0 แปลว่าไม่มีข้อมูลในคิว
2. ถ้ามีข้อมูลในคิวจะทำการคัดลอกข้อมูลที่ตำแหน่ง front ไว้ : $item = Q[front]$

ขั้นตอนวิธีการนำข้อมูลเข้าและออกจากคิววงกลม

3. ปรับค่า front เลื่อนออกไป 1 ตำแหน่ง โดยในการปรับตำแหน่งจะต้องตรวจสอบว่า front อยู่ตำแหน่งสุดท้ายแล้วหรือยัง

3.1 ถ้าอยู่ตำแหน่งสุดท้าย front จะกลายเป็นตำแหน่งแรก

3.2 ถ้าไม่ได้อยู่ตำแหน่งสุดท้าย $\text{front} = \text{front} + 1$

4. ส่งค่าที่คัดลอกไว้กลับไป : return item