

# Concurrent Queue

---

Concurrent Programming

# Introduction

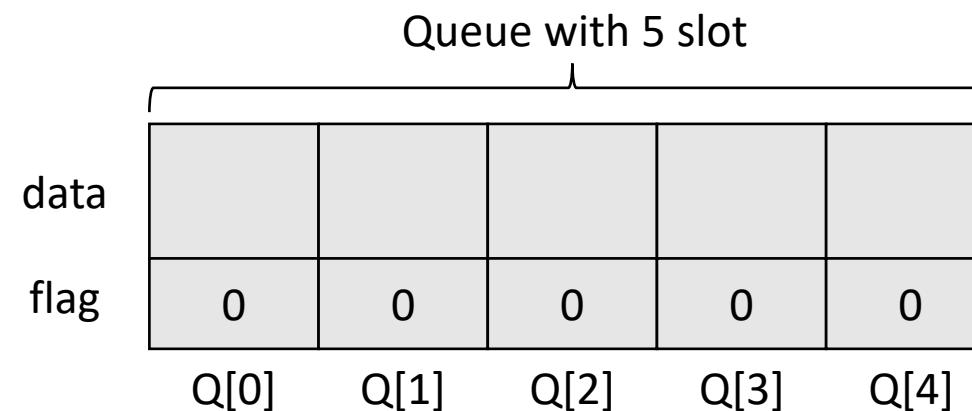
---

- Bounded Lock-free Queue
- Evaluation
  - Queue with coarse-grained locking
  - Unbounded Lock-free Queue (covered in lecture note)
  - Bounded Lock-free Queue

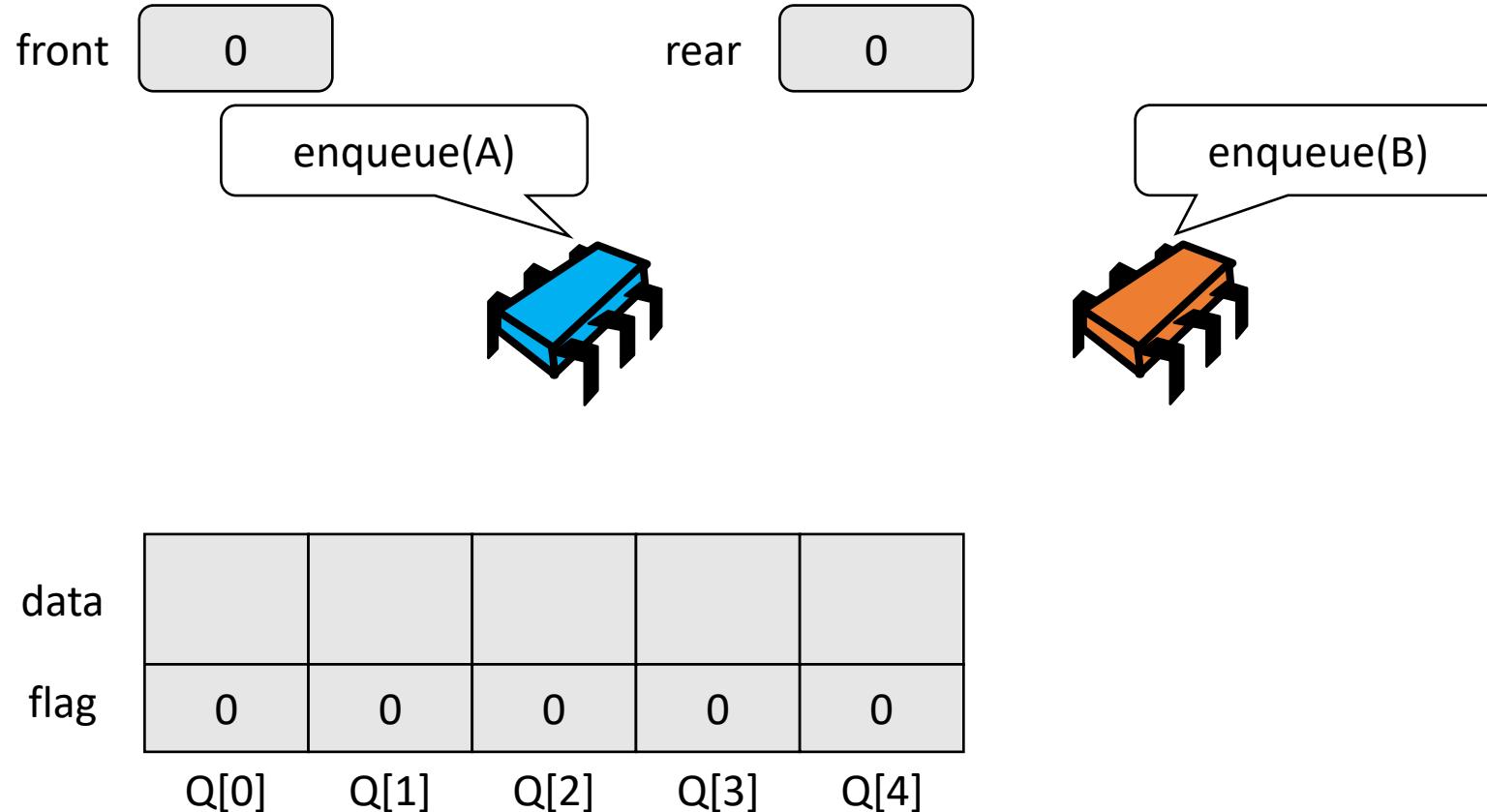
# Bounded lock-free queue

front 0

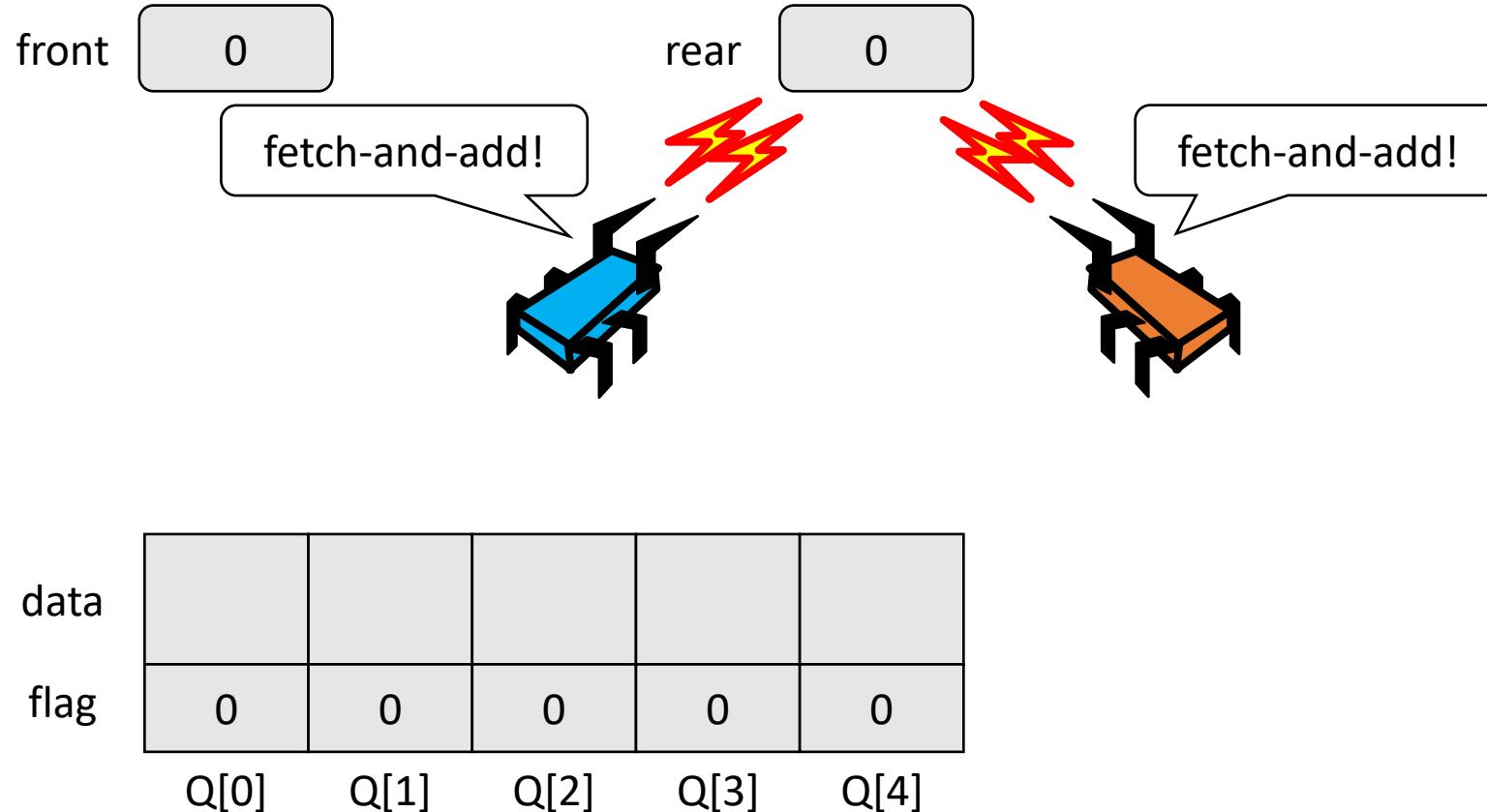
rear 0



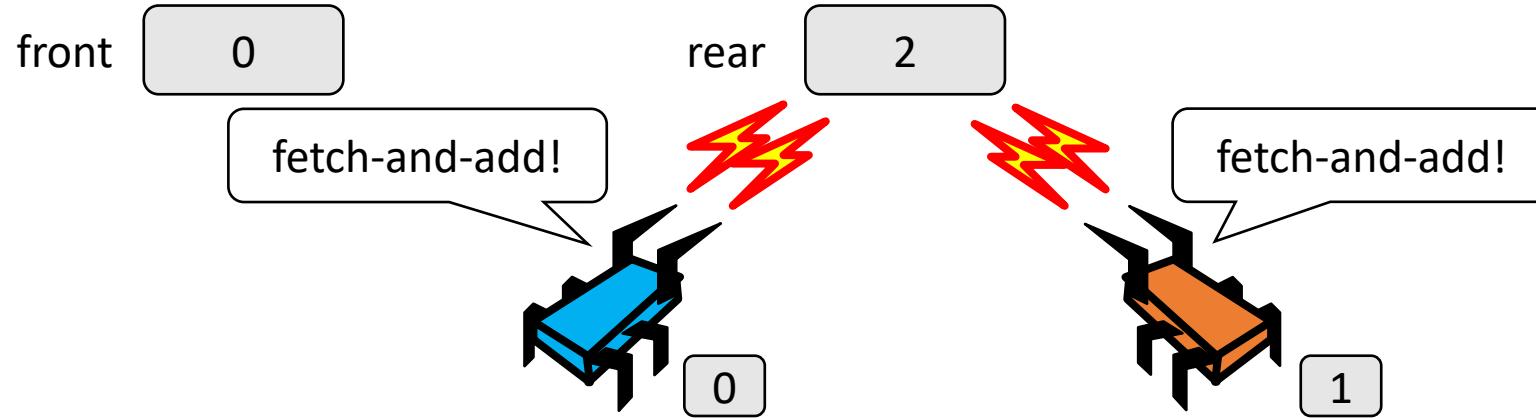
# Enqueue



# Enqueue

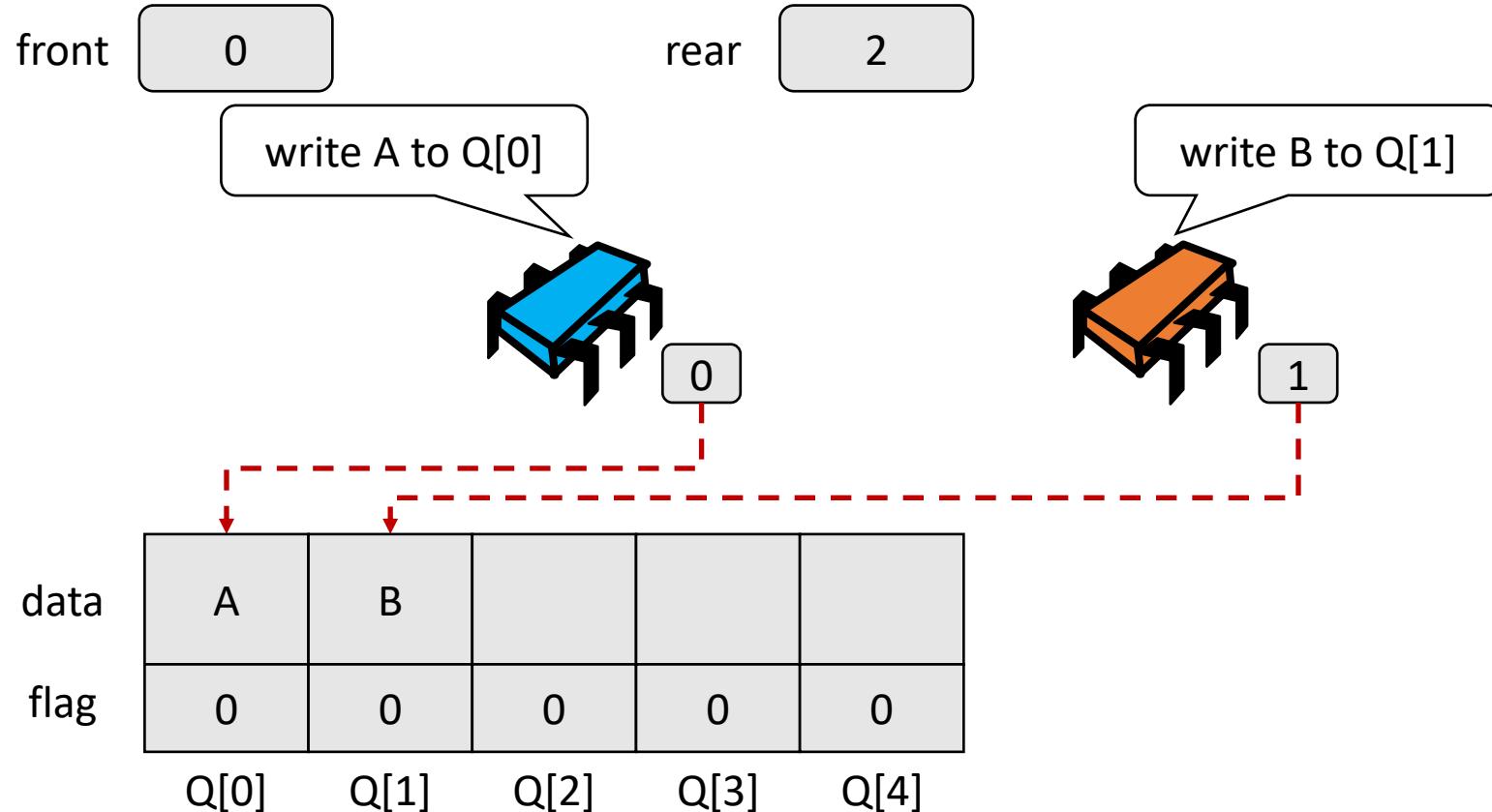


# Enqueue

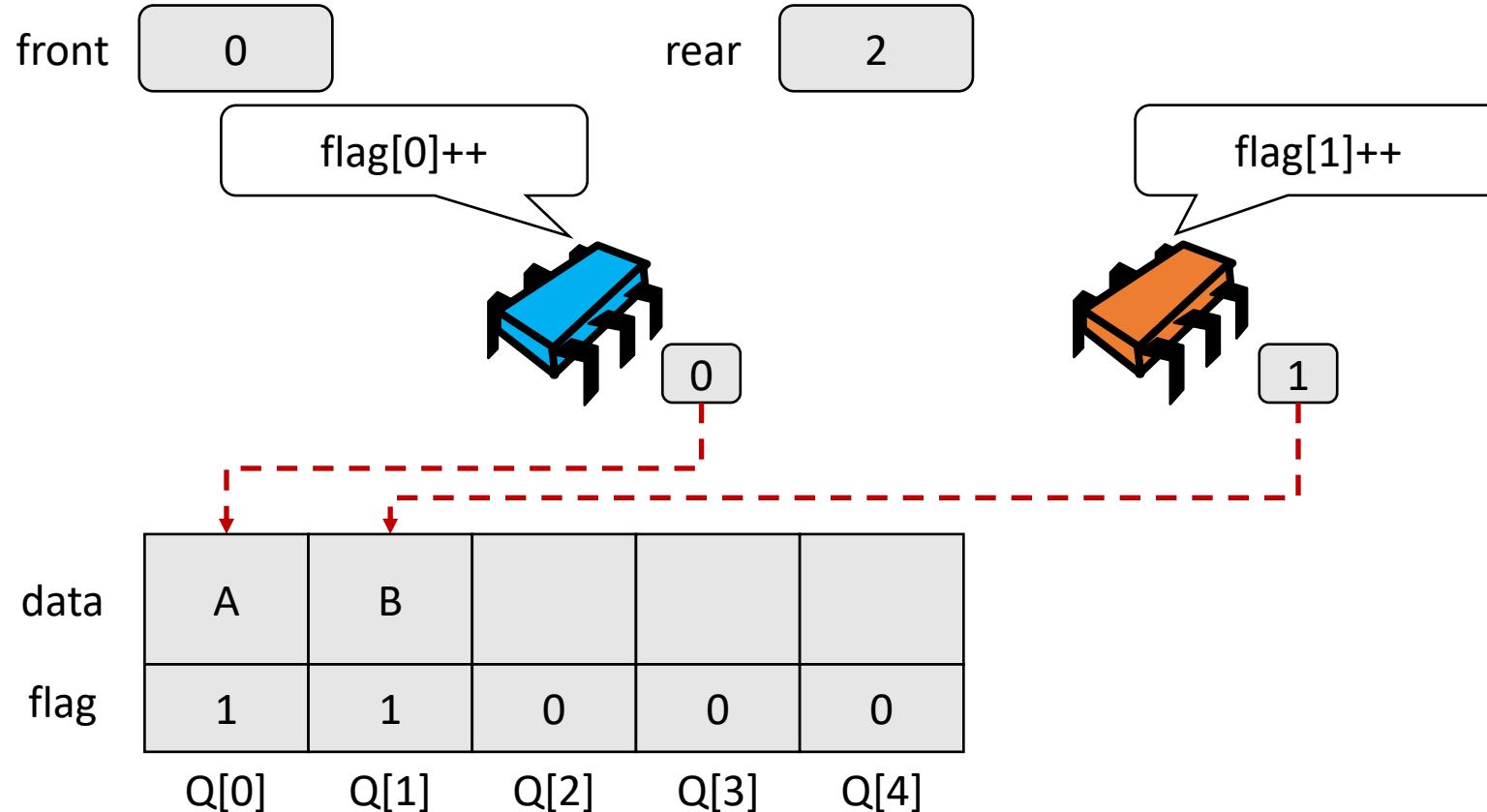


data					
flag	0	0	0	0	0
	<code>Q[0]</code>	<code>Q[1]</code>	<code>Q[2]</code>	<code>Q[3]</code>	<code>Q[4]</code>

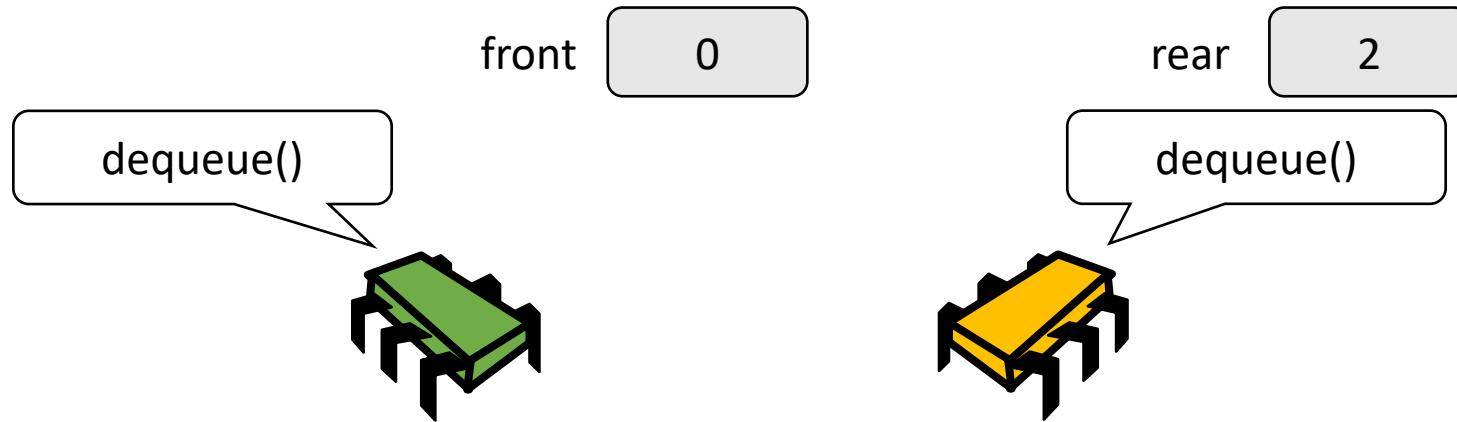
# Enqueue



# Enqueue

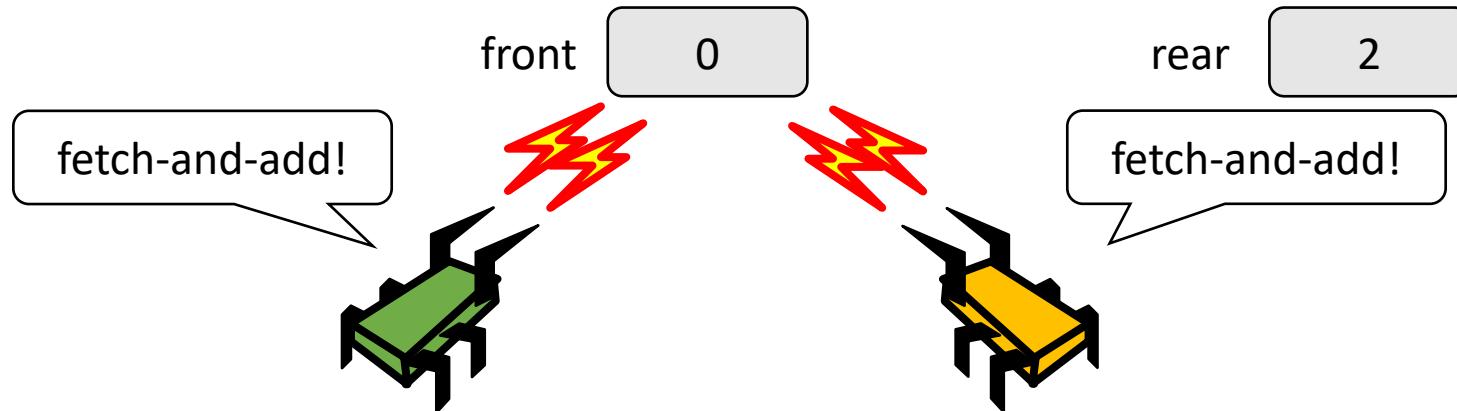


# Dequeue



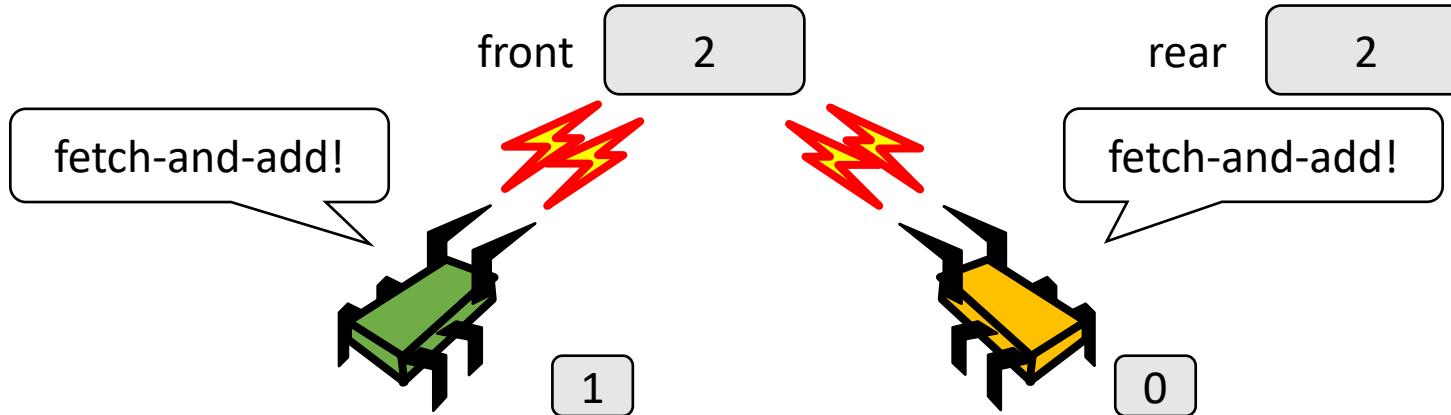
data	A	B			
flag	1	1	0	0	0
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]

# Dequeue



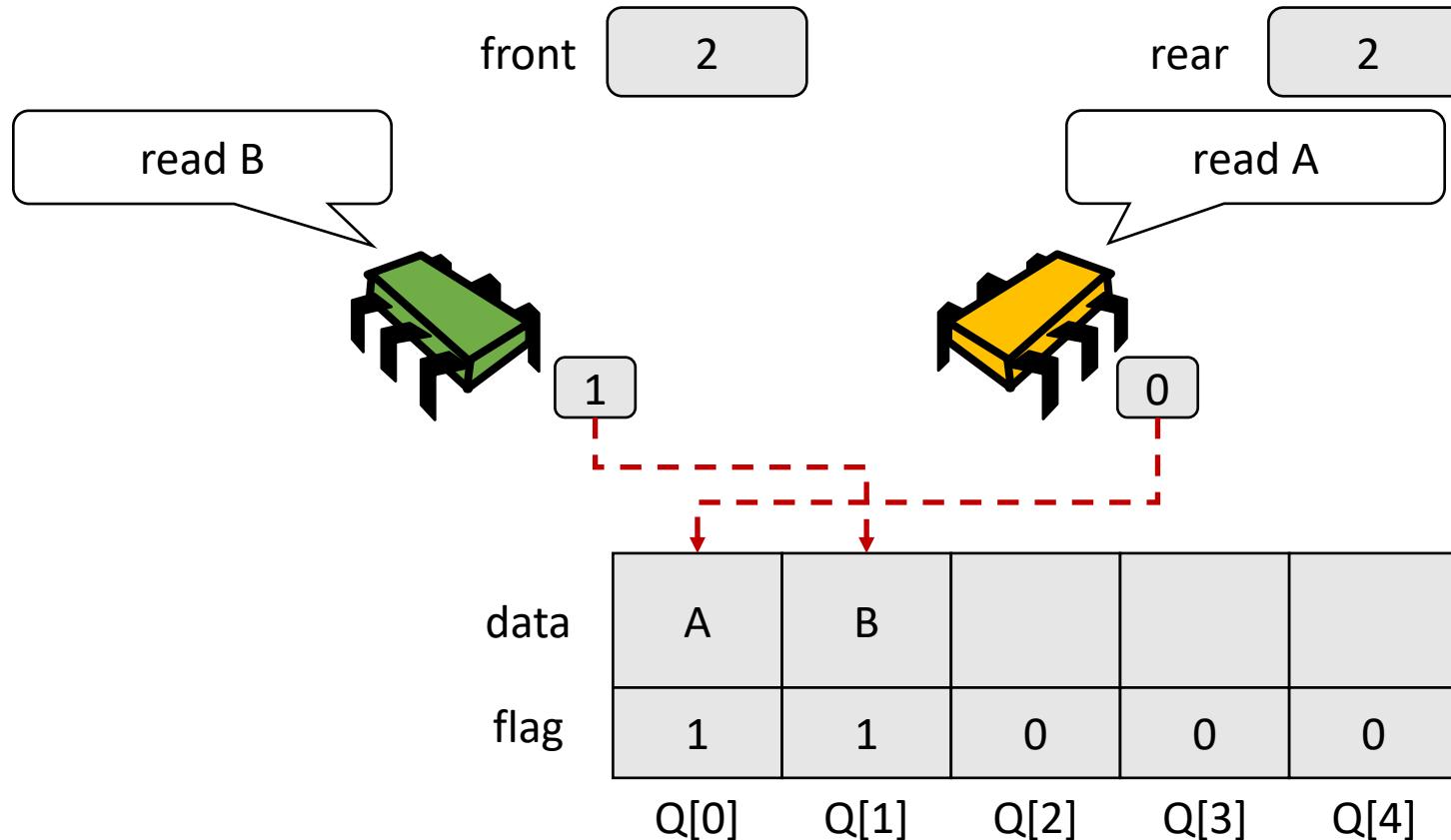
data	A	B			
flag	1	1	0	0	0
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]

# Dequeue

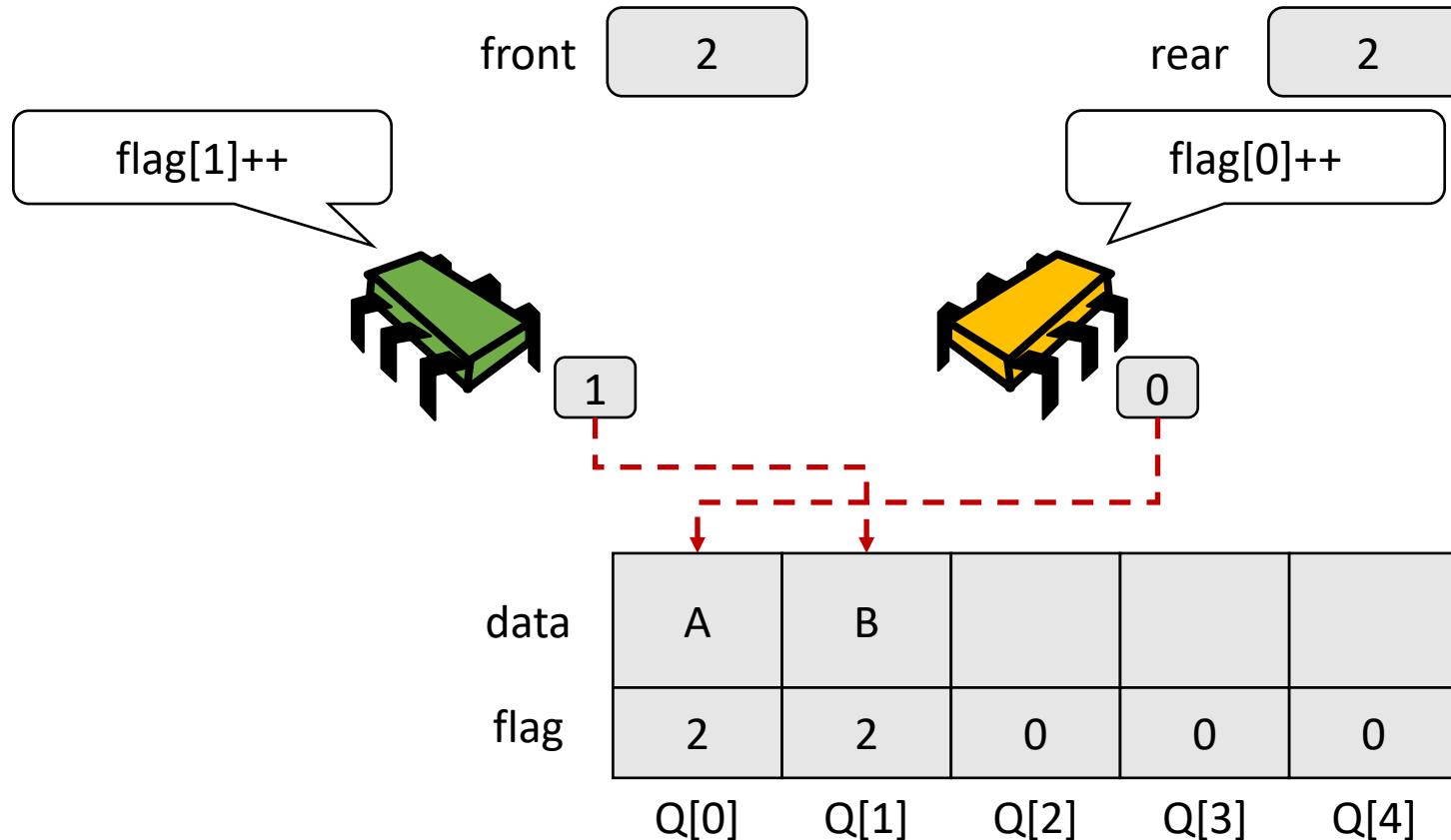


data	A	B			
flag	1	1	0	0	0
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]

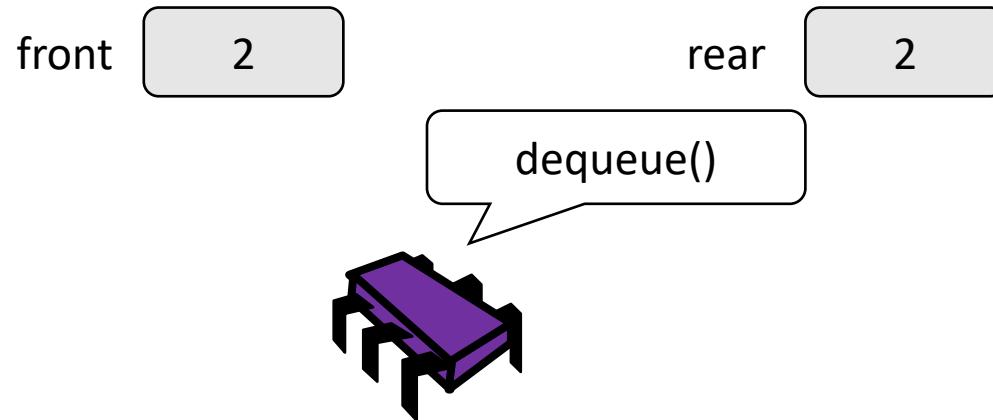
# Dequeue



# Dequeue

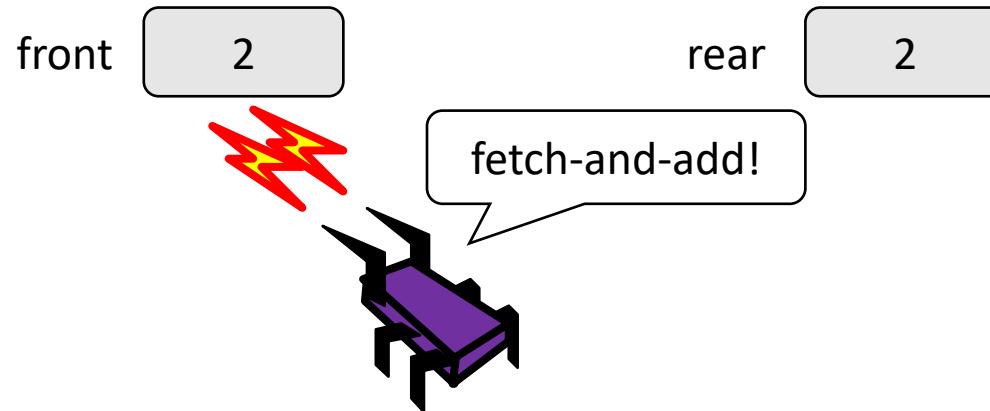


# Dequeue



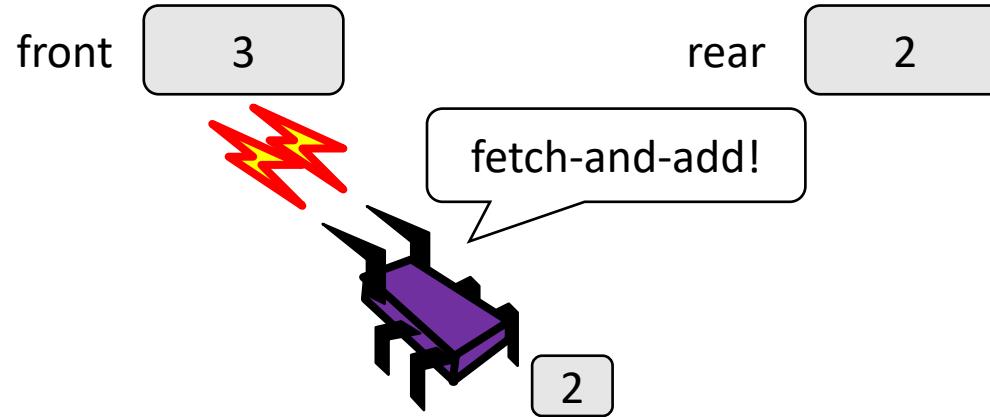
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]
data	A	B			
flag	2	2	0	0	0

# Dequeue



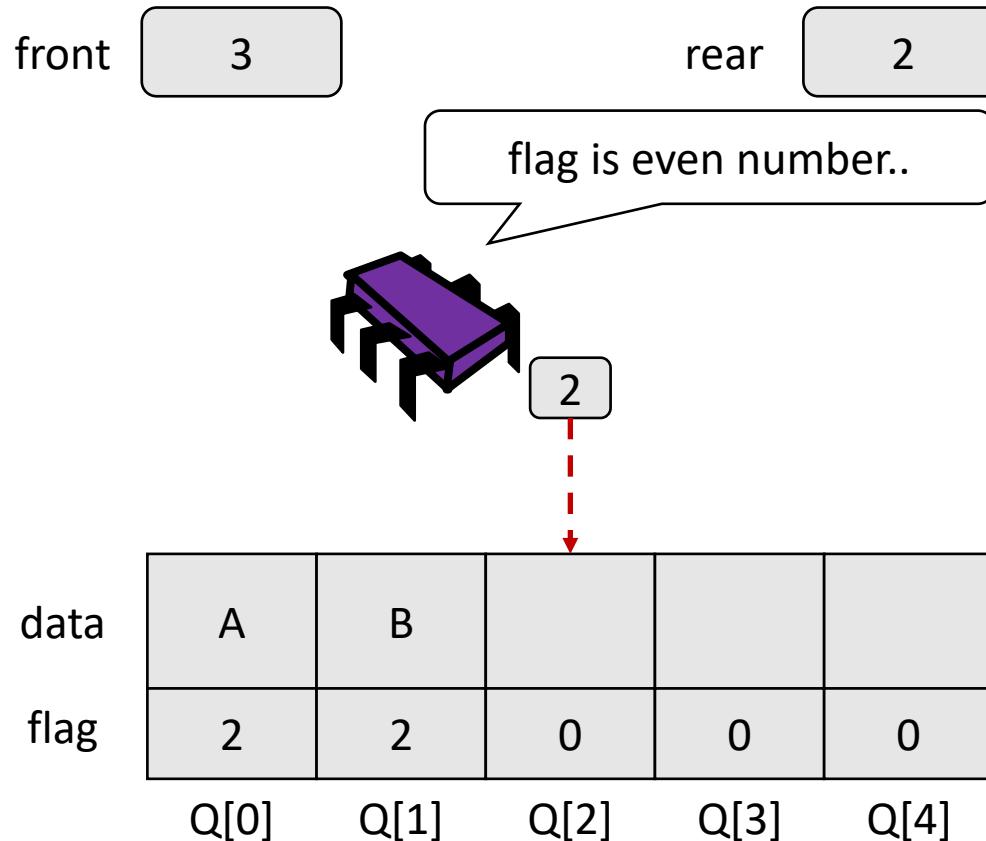
data	A	B			
flag	2	2	0	0	0
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]

# Dequeue

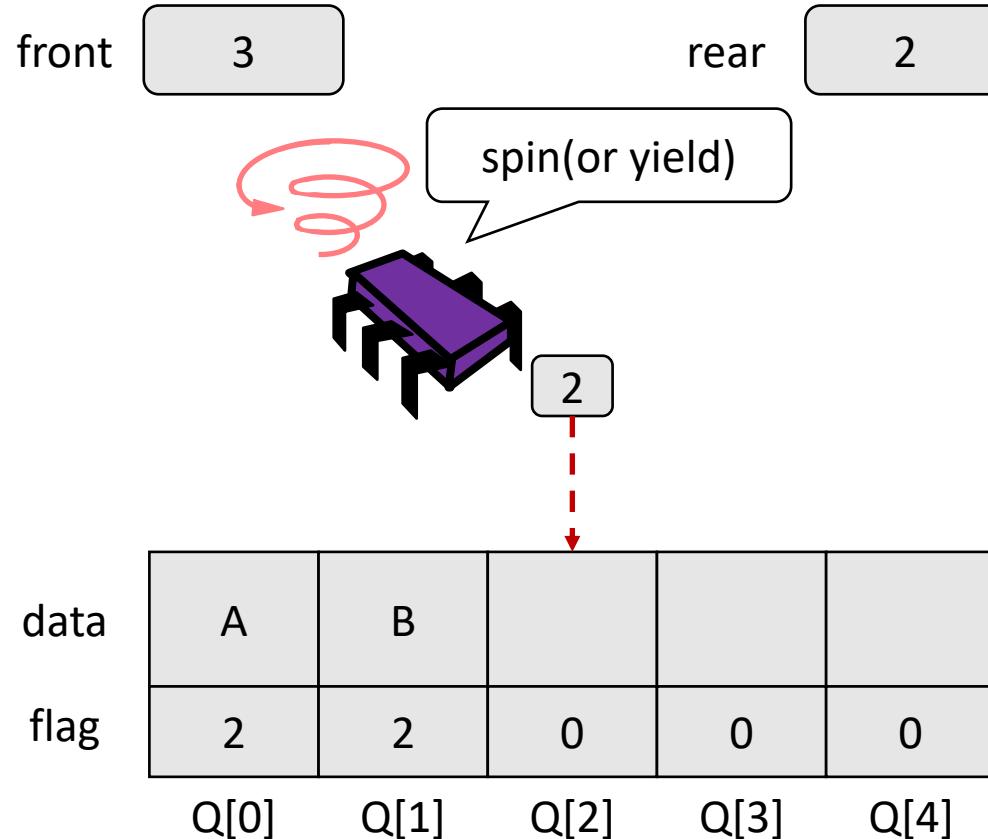


data	A	B			
flag	2	2	0	0	0
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]

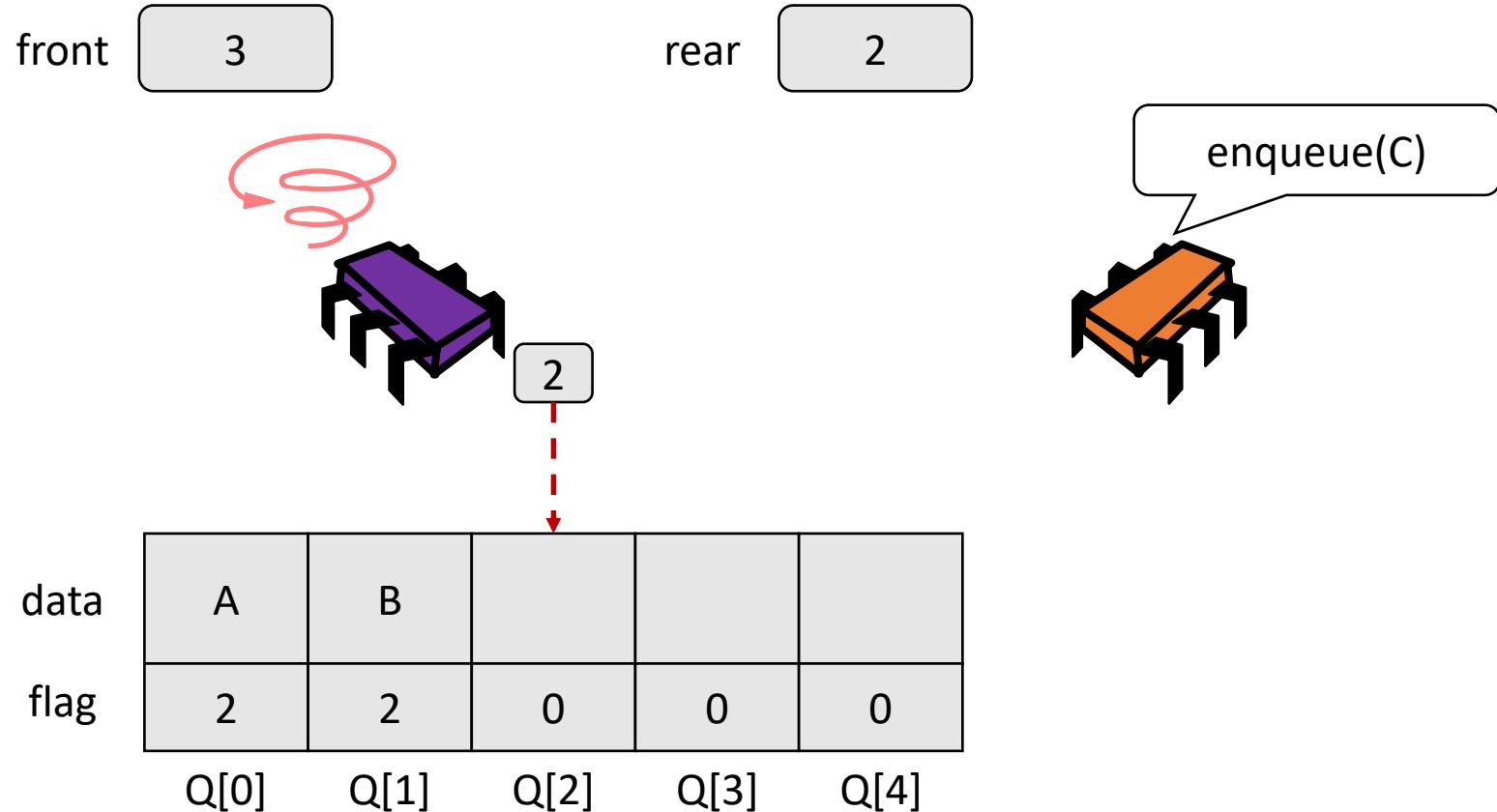
# Dequeue



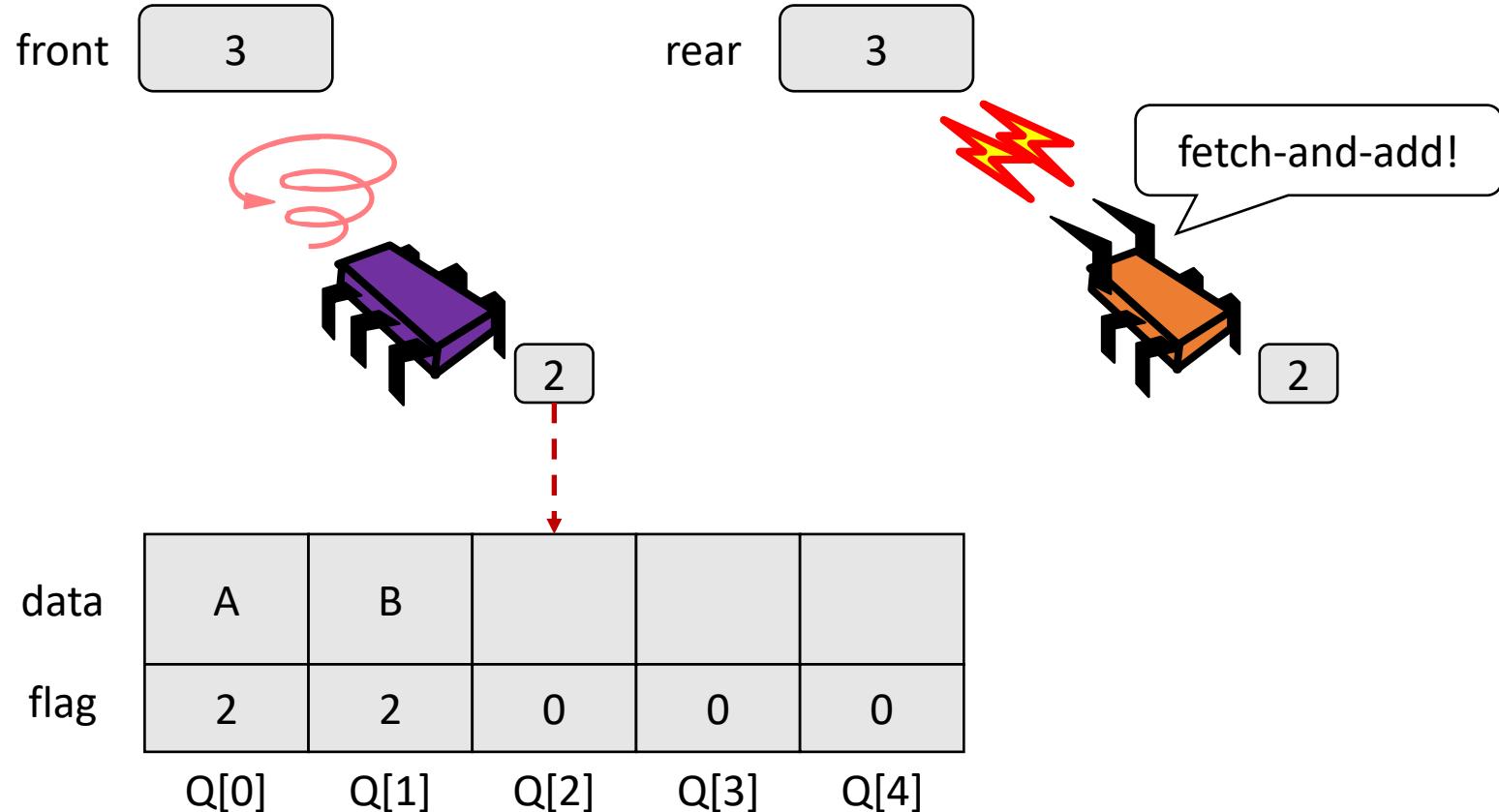
# Dequeue



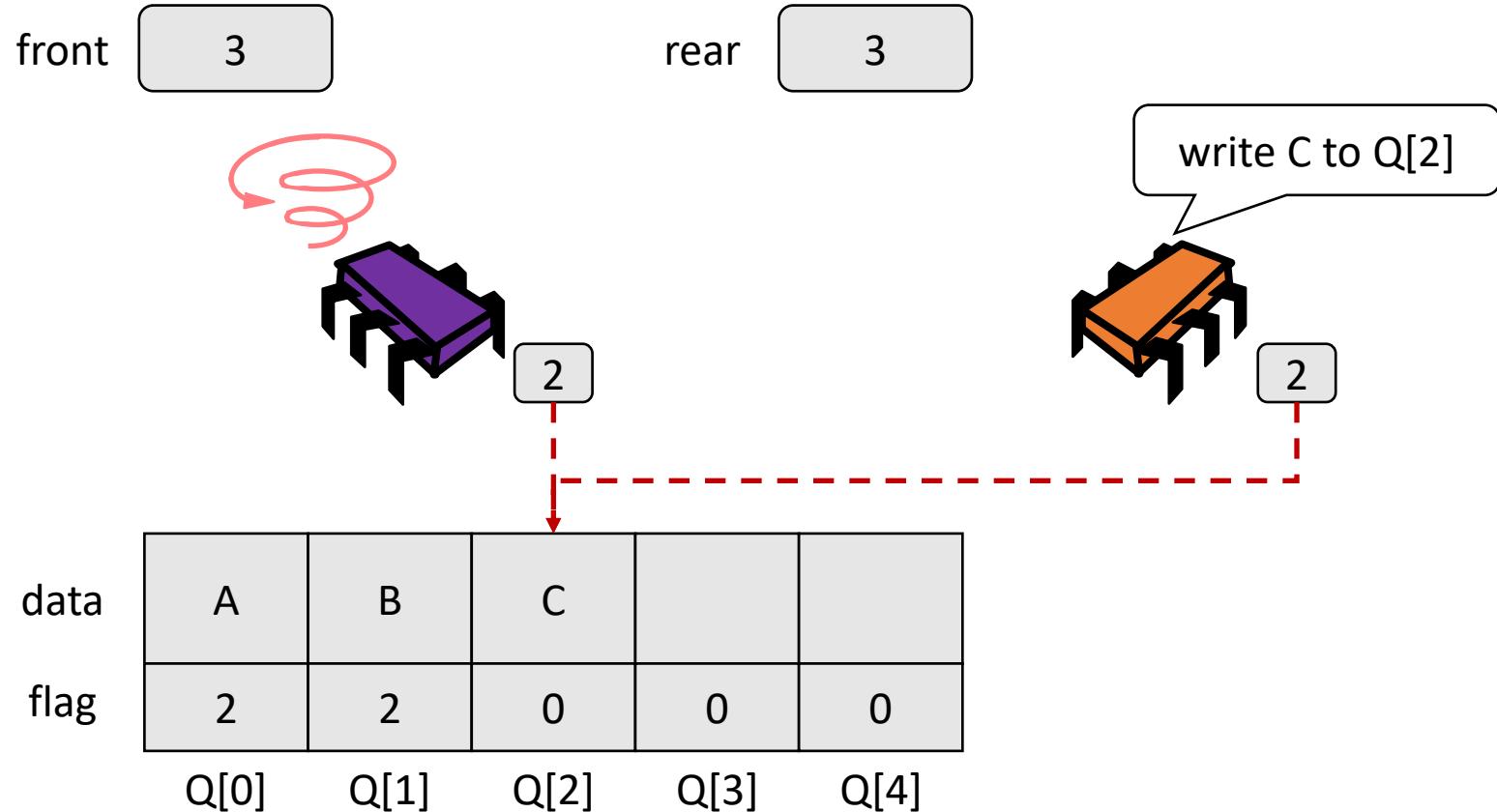
# Dequeue



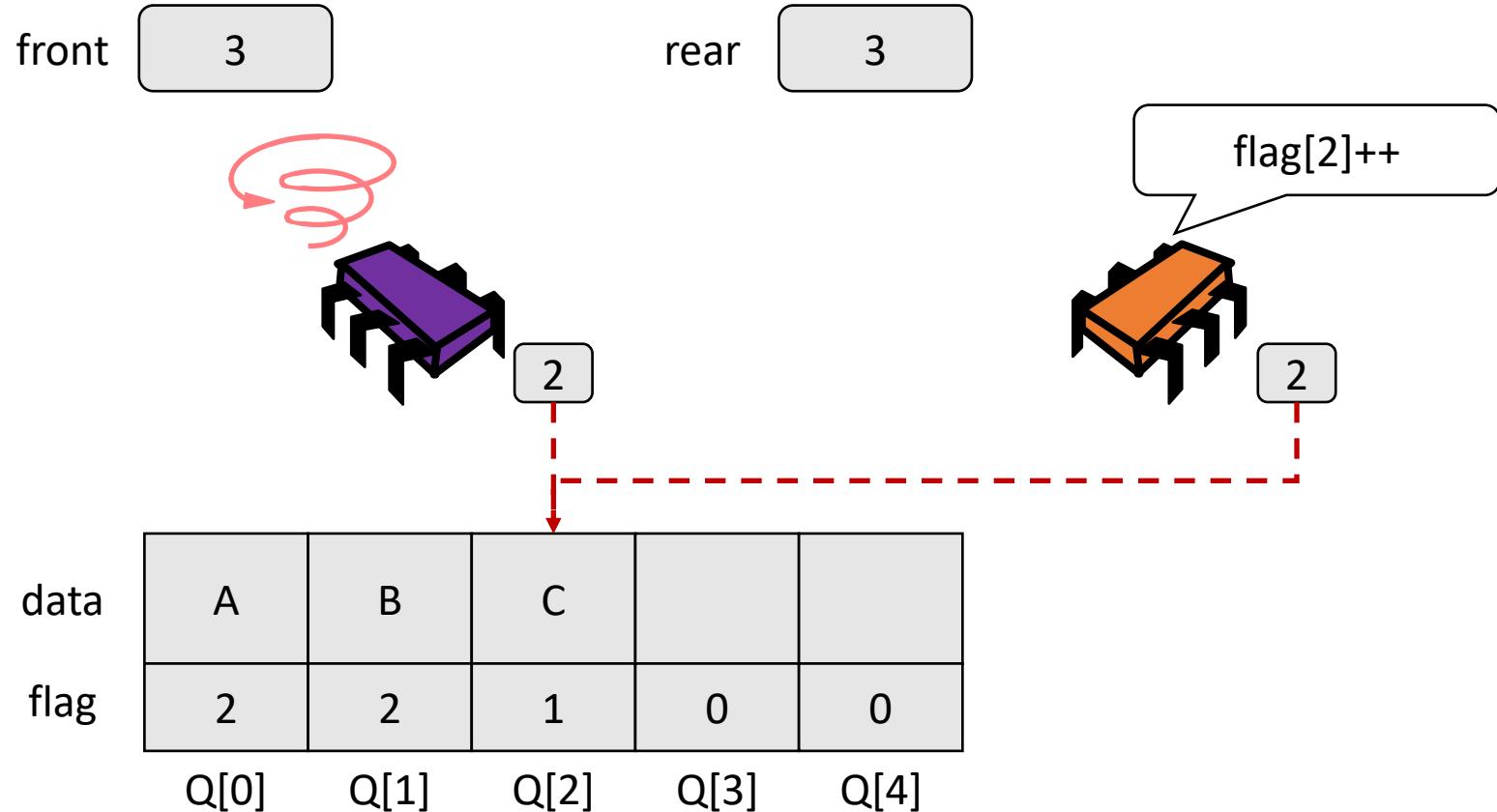
# Dequeue



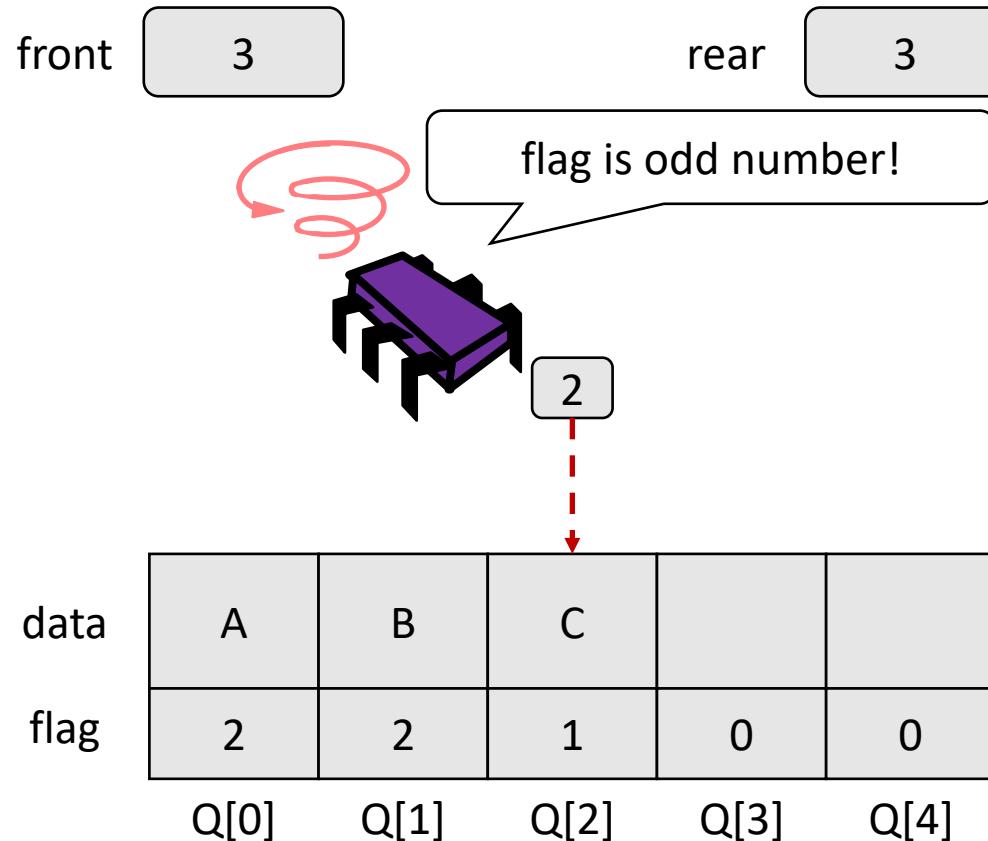
# Dequeue



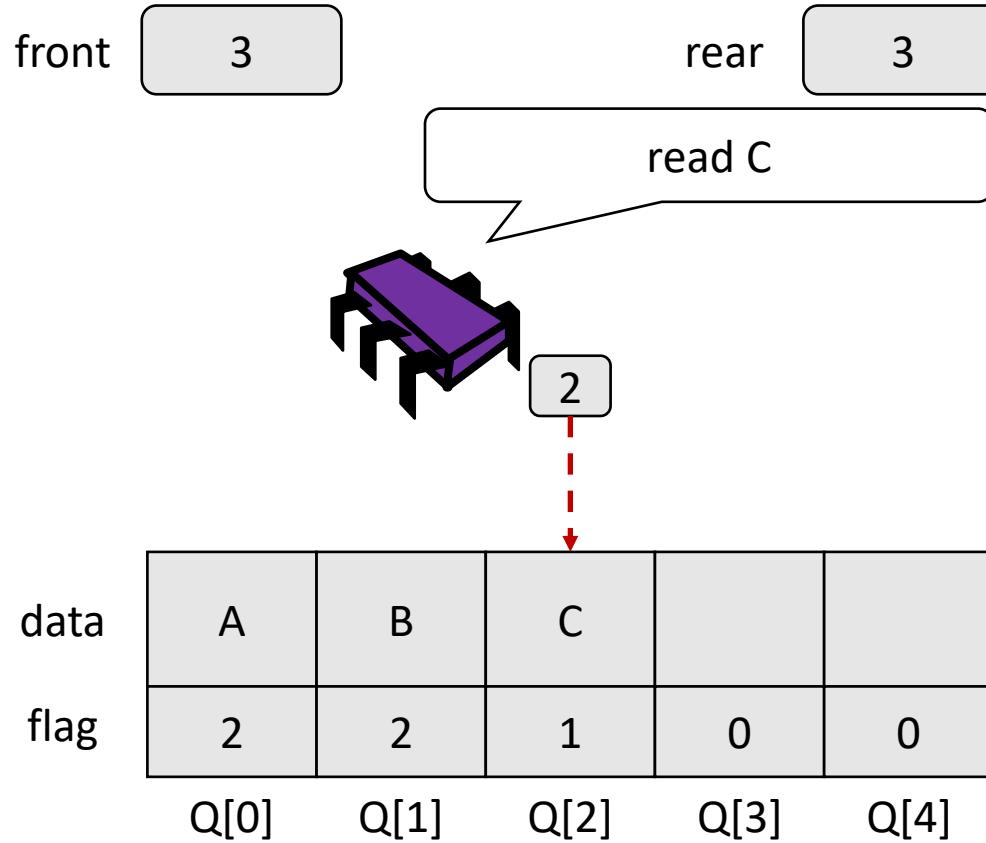
# Dequeue



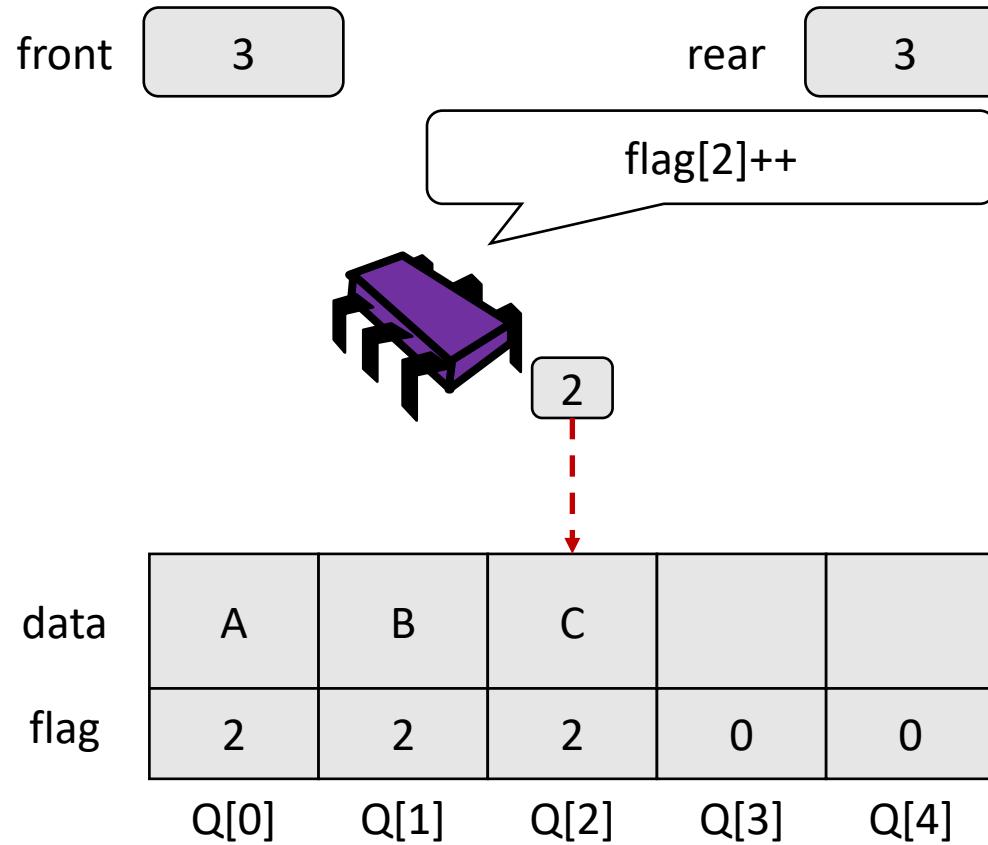
# Dequeue



# Deque



# Dequeue



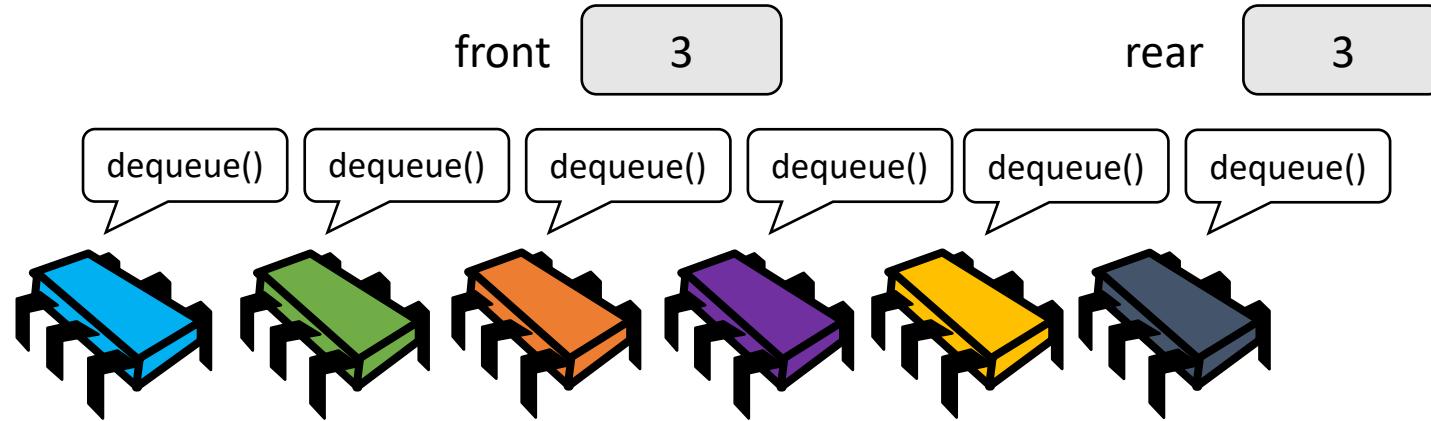
# Dequeue

front 3

rear 3

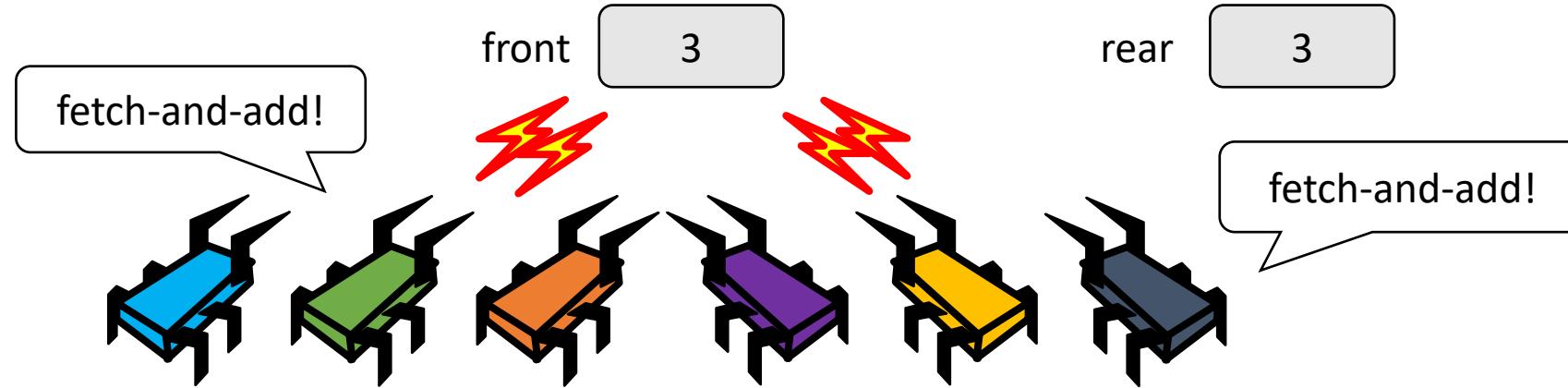
data	A	B	C		
flag	2	2	2	0	0
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]

# Dequeue



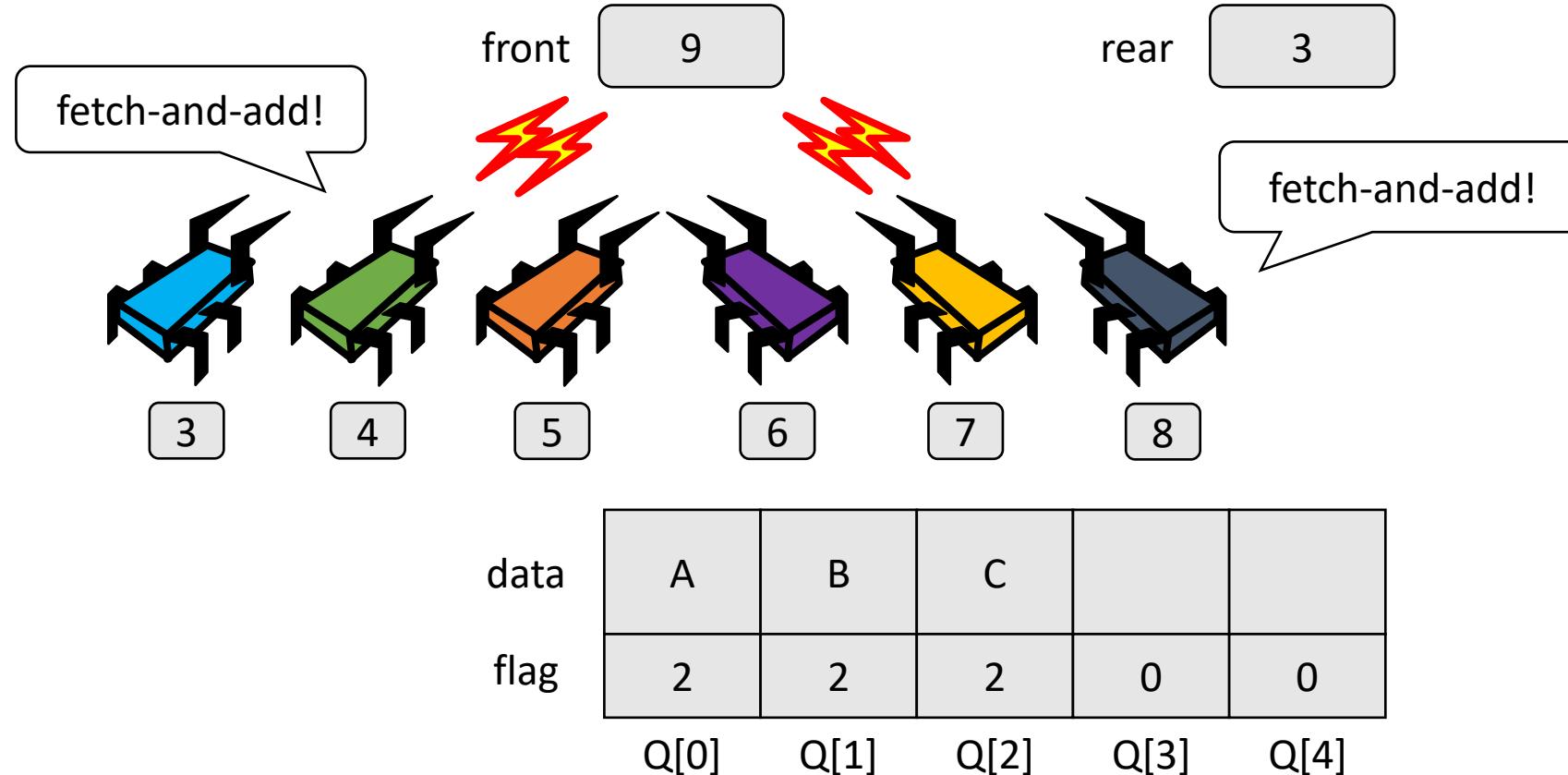
data	A	B	C		
flag	2	2	2	0	0
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]

# Dequeue

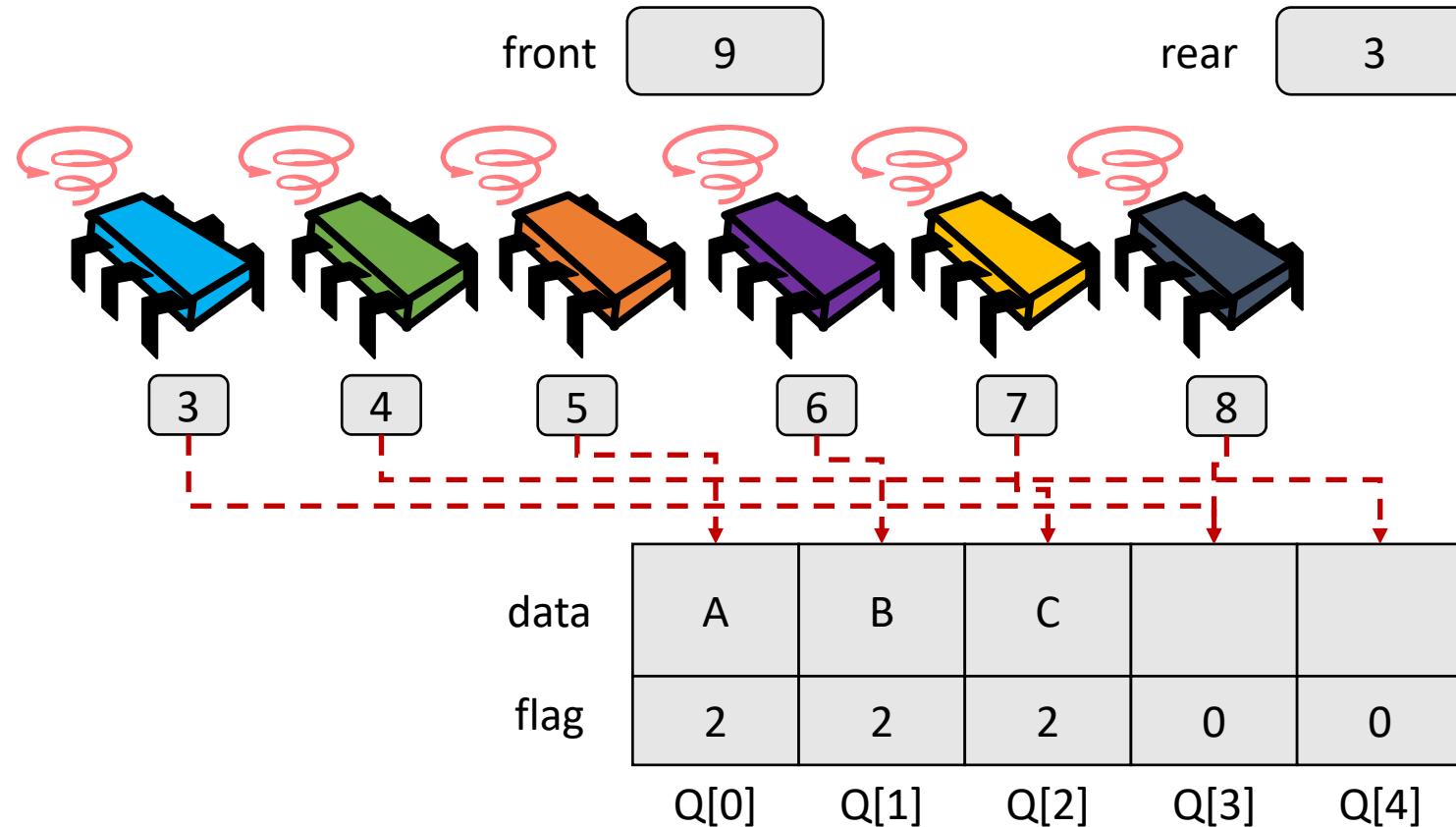


data	A	B	C		
flag	2	2	2	0	0
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]

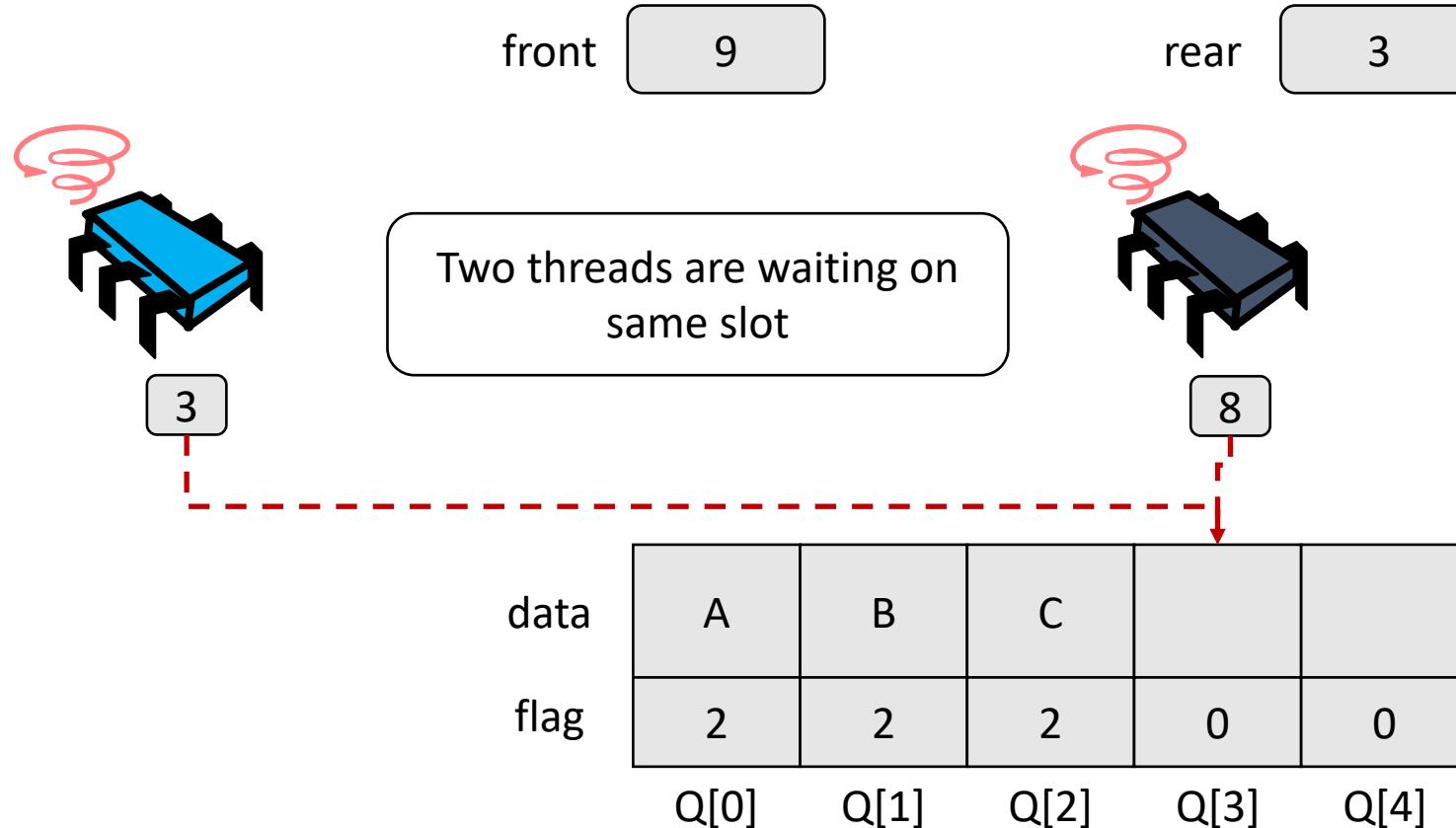
# Dequeue



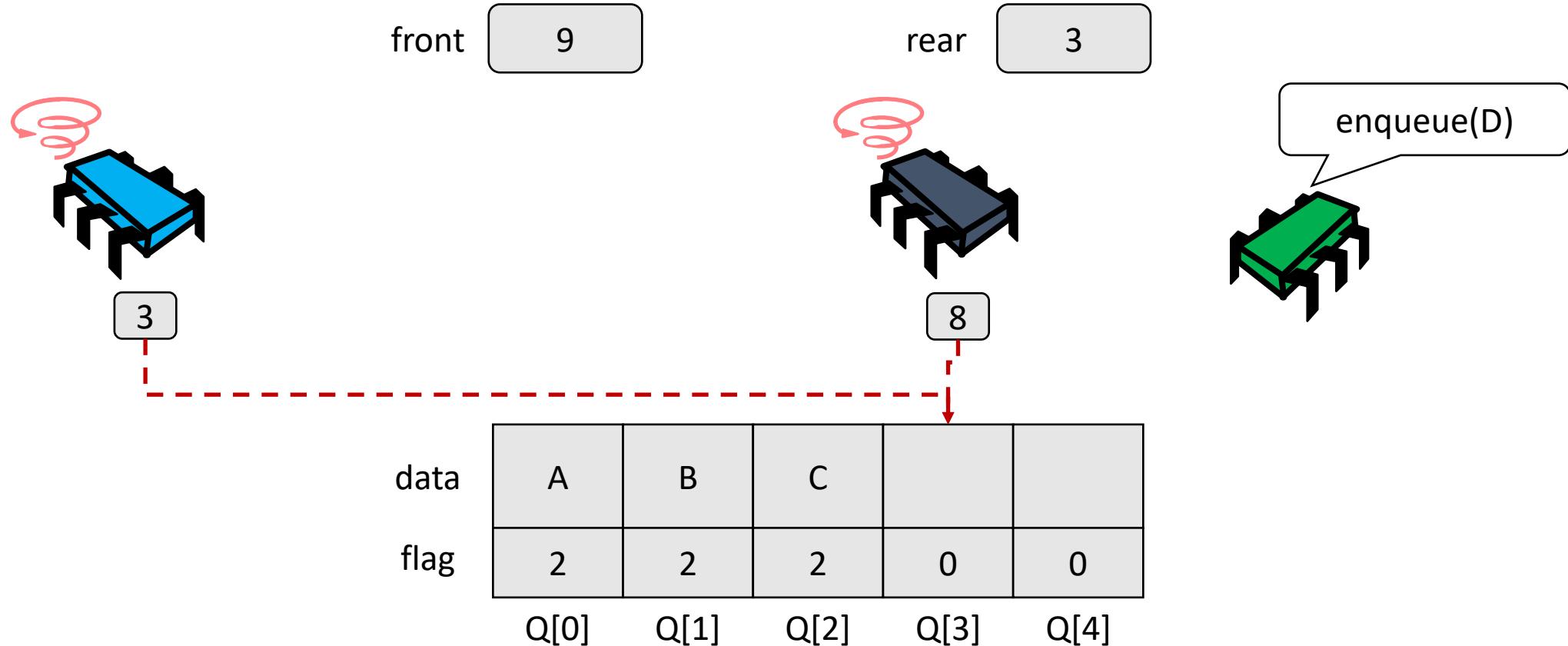
# Dequeue



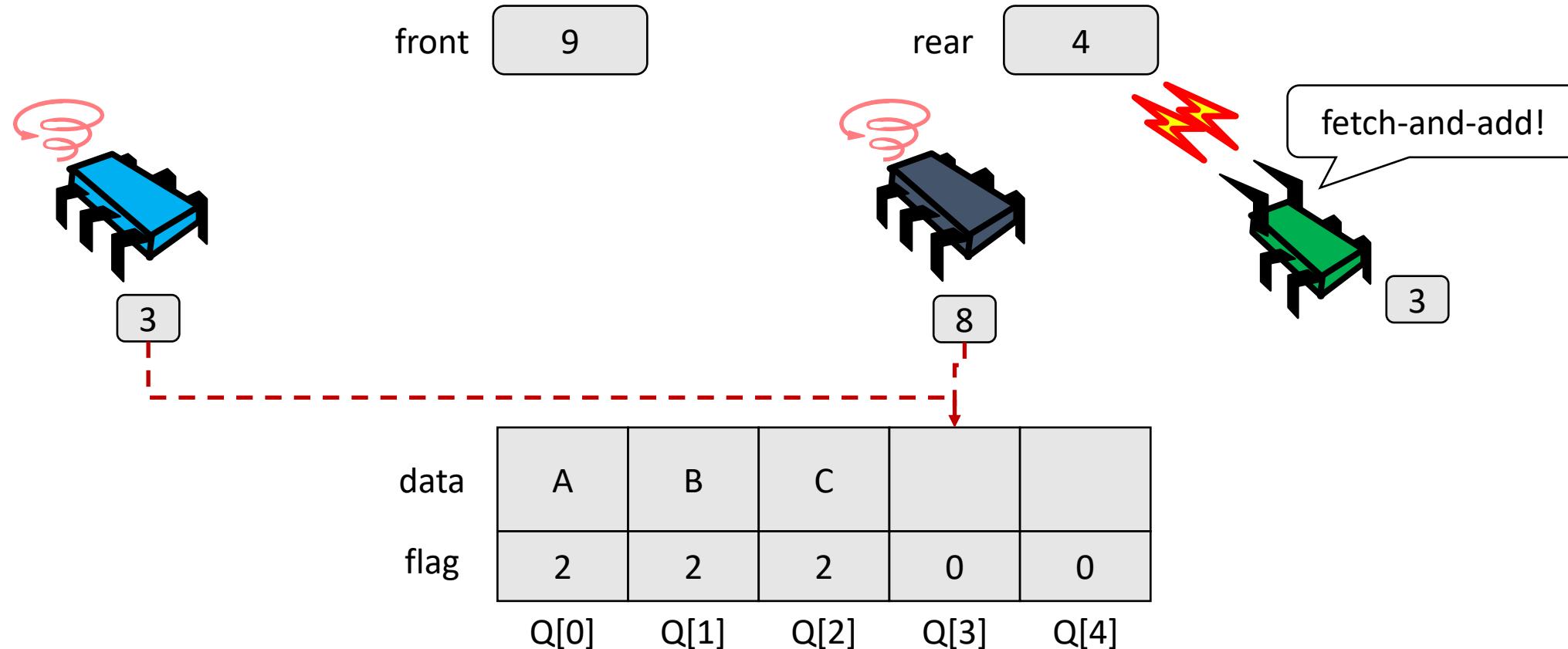
# Dequeue



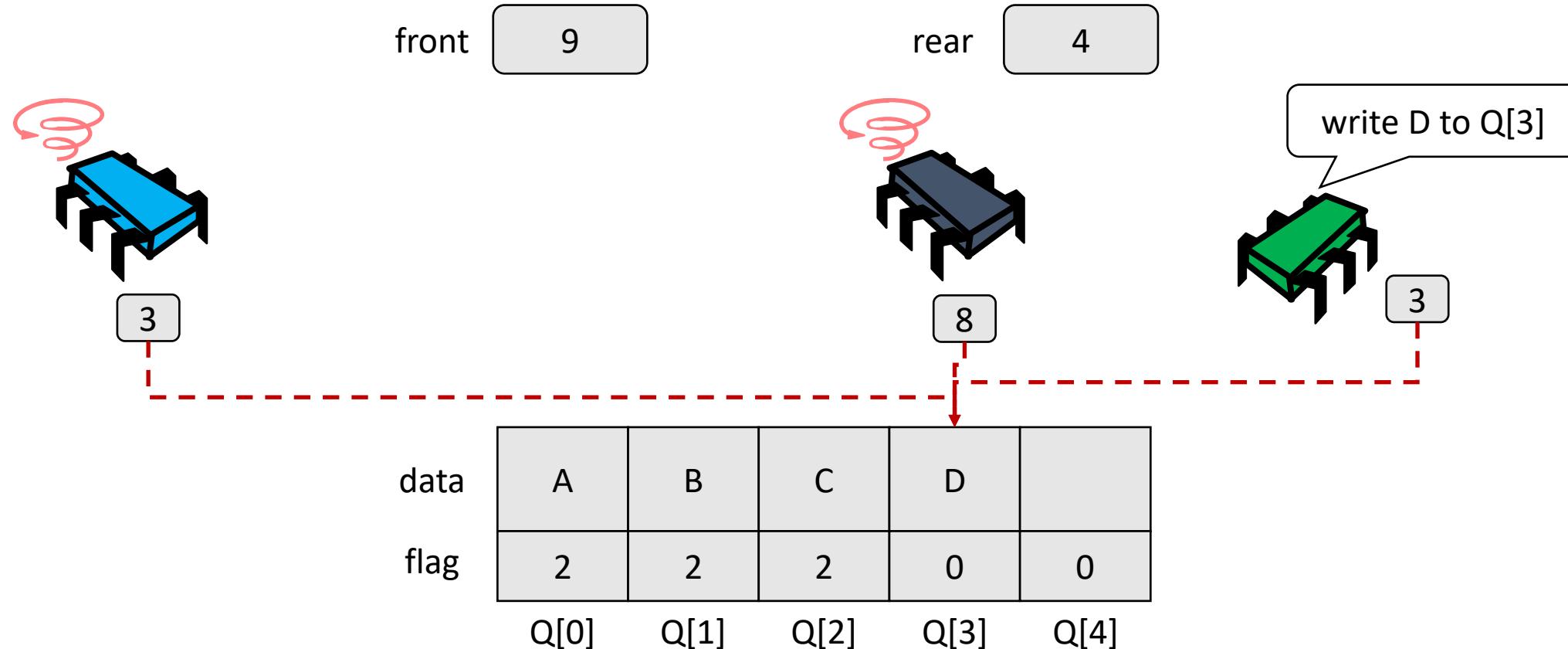
# Dequeue



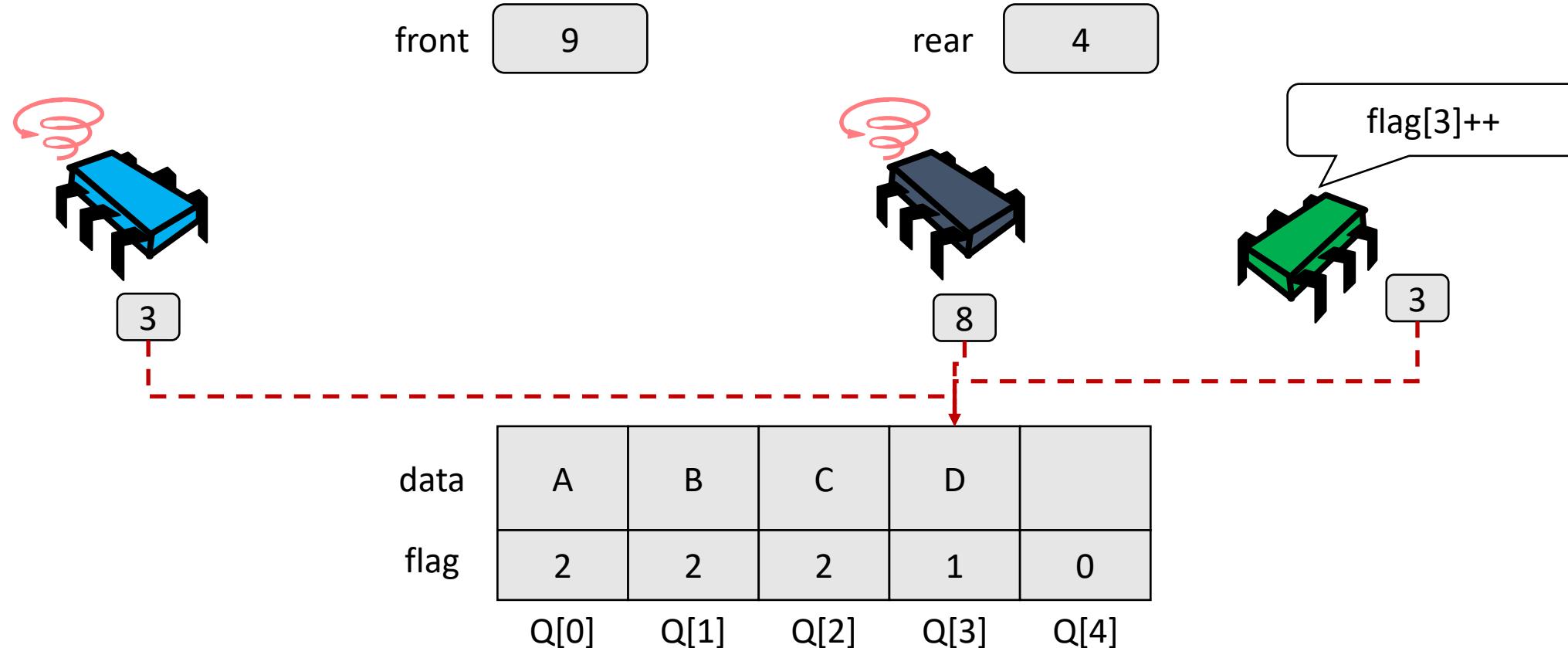
# Dequeue



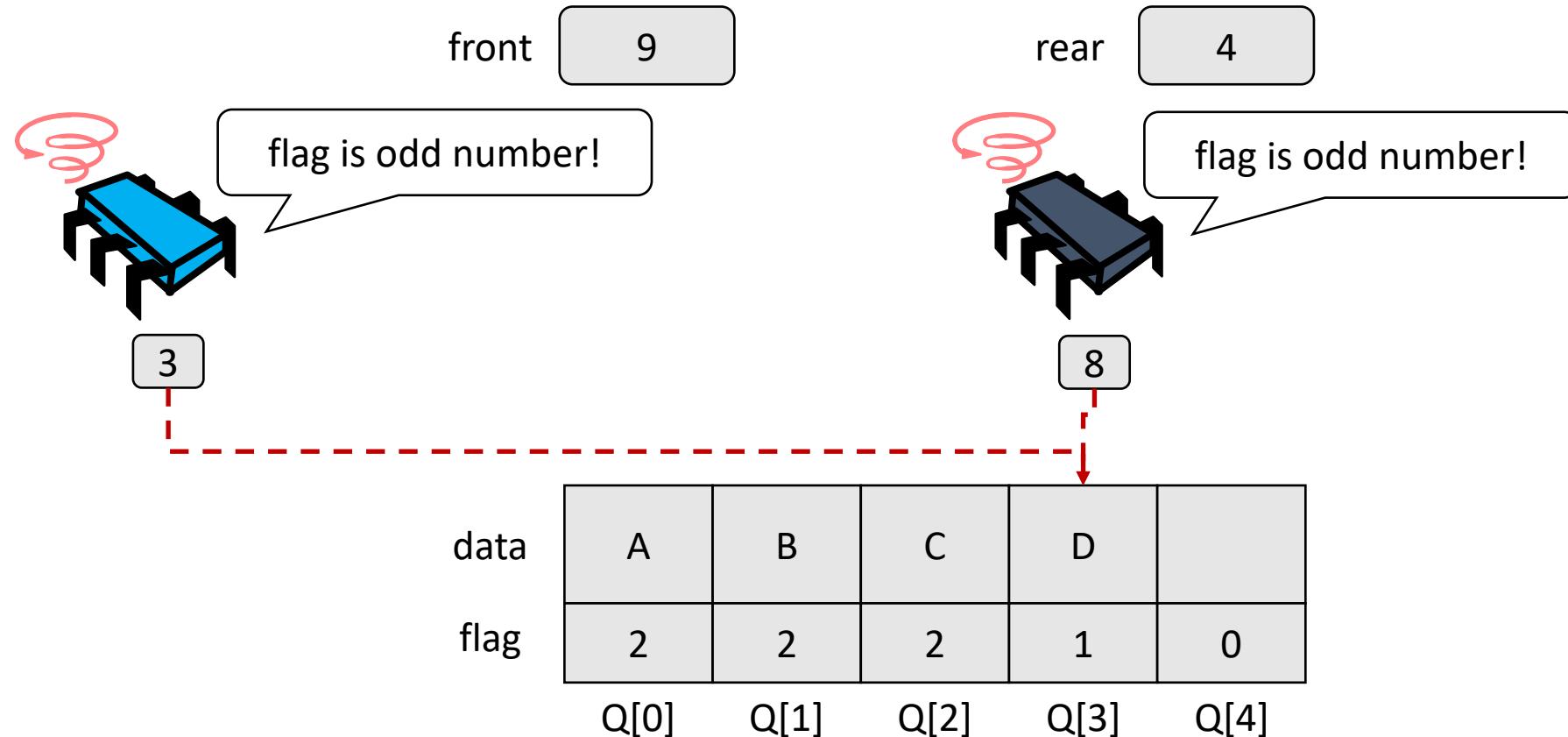
# Dequeue



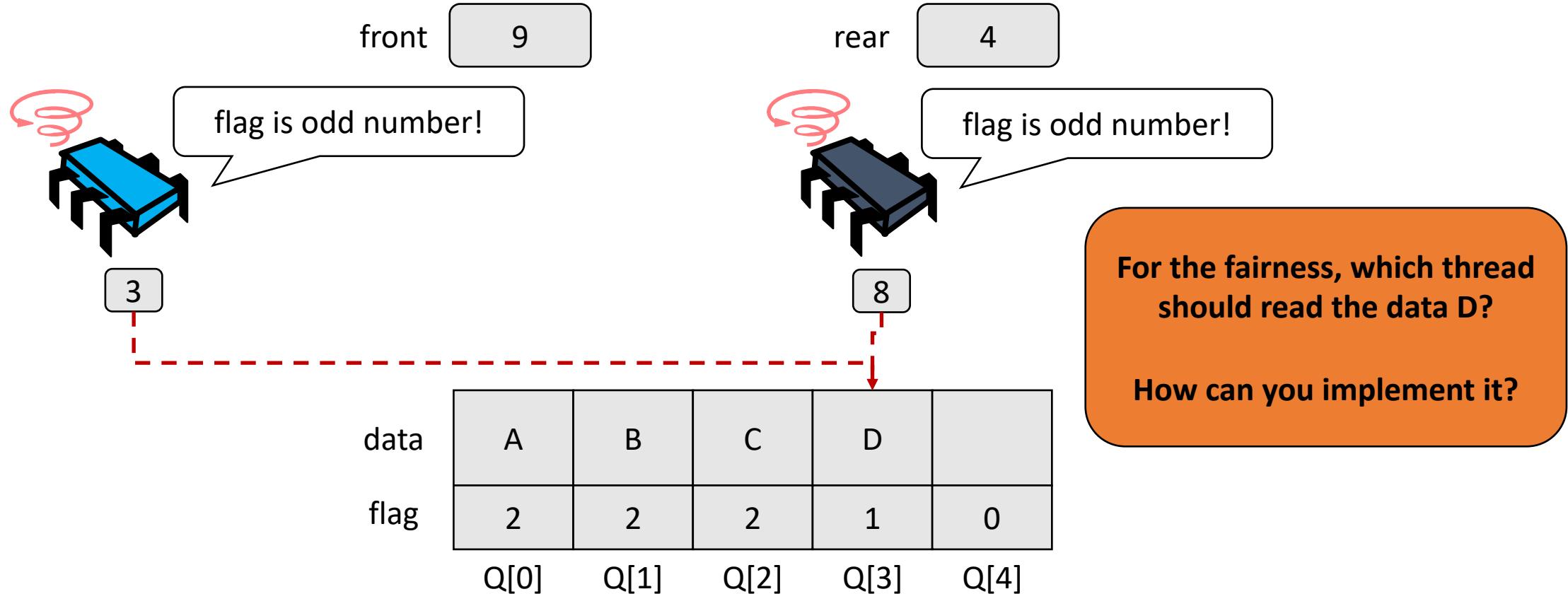
# Dequeue



# Dequeue



# Dequeue



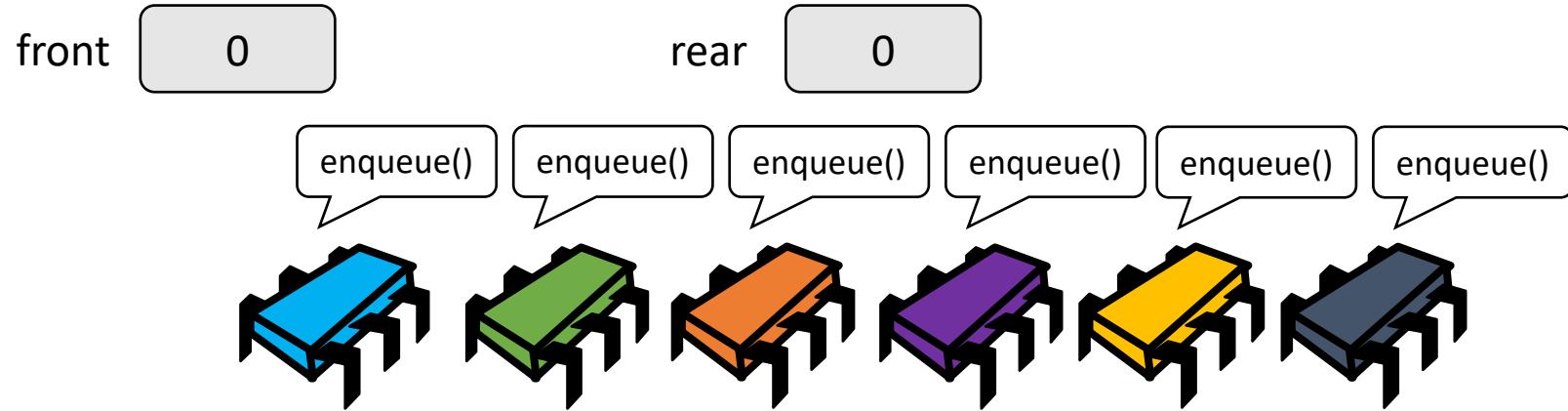
# Enqueue

front 0

rear 0

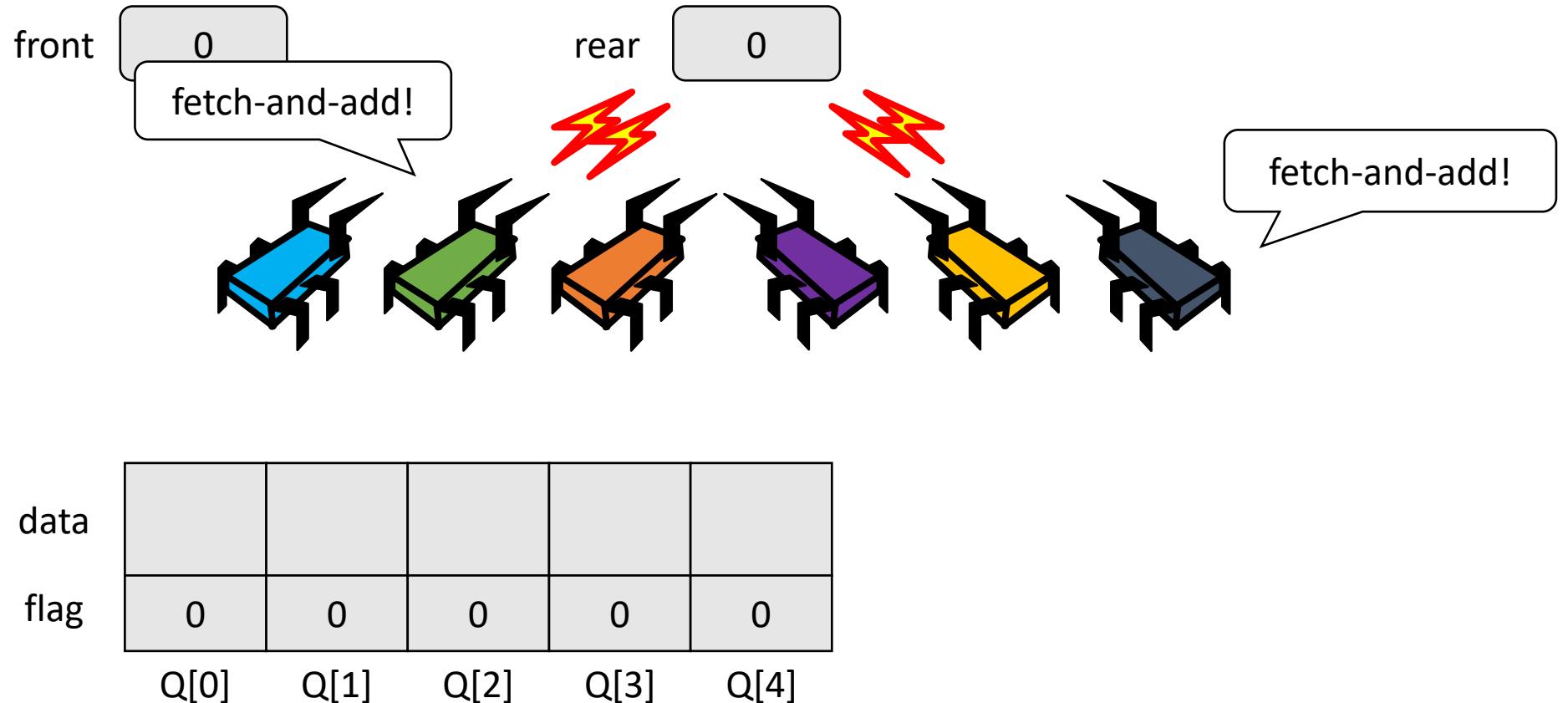
data					
flag	0	0	0	0	0
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]

# Enqueue

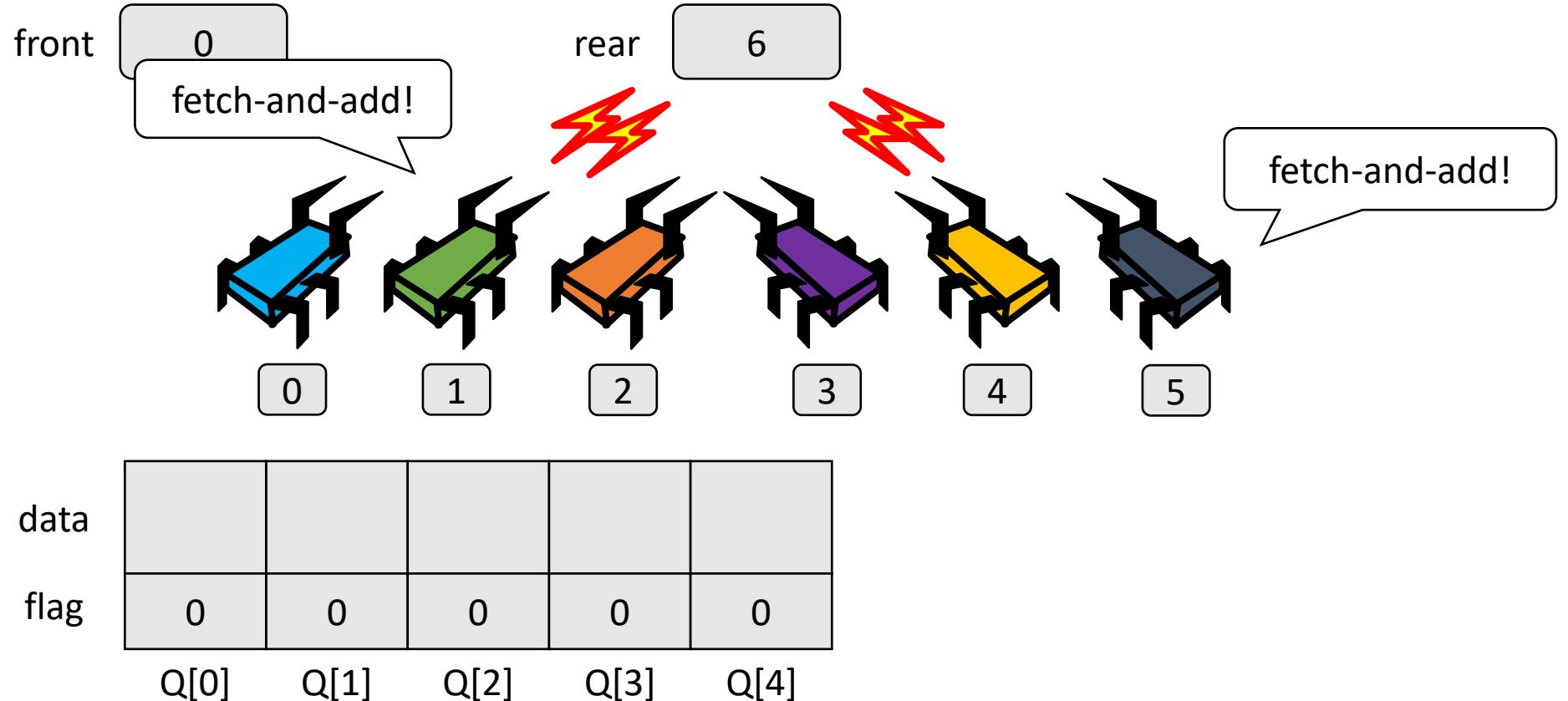


data					
flag	0	0	0	0	0
	Q[0]	Q[1]	Q[2]	Q[3]	Q[4]

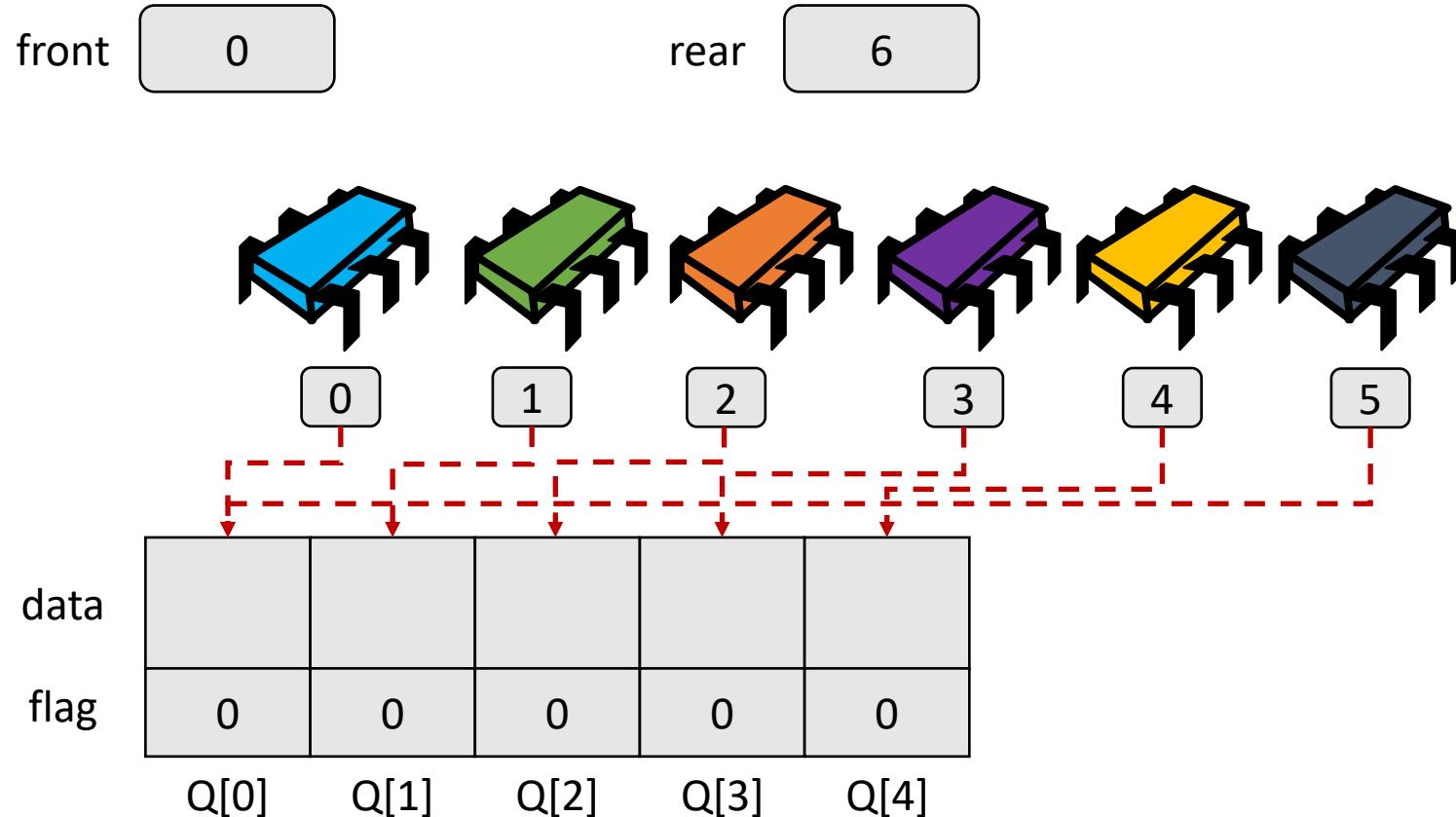
# Enqueue



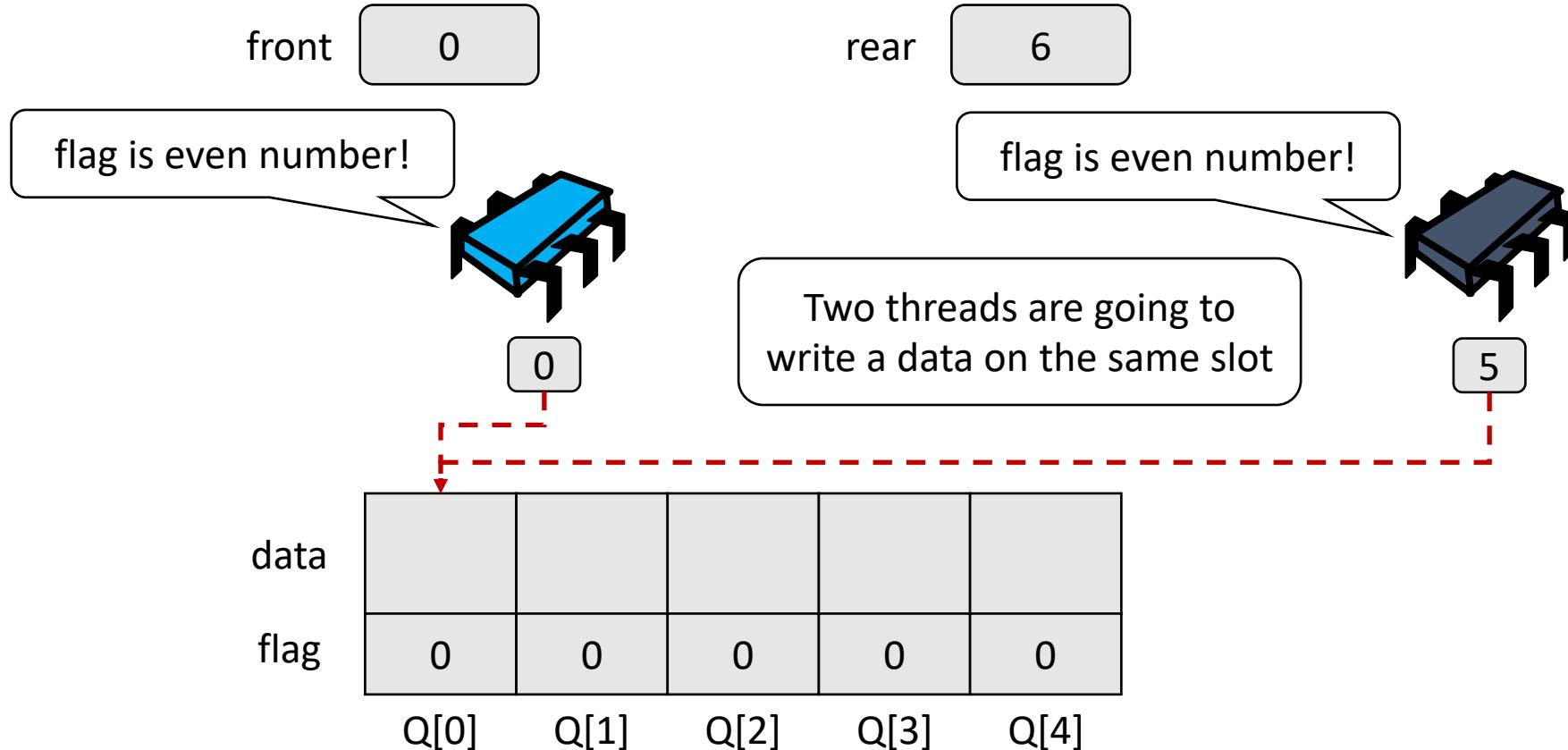
# Enqueue



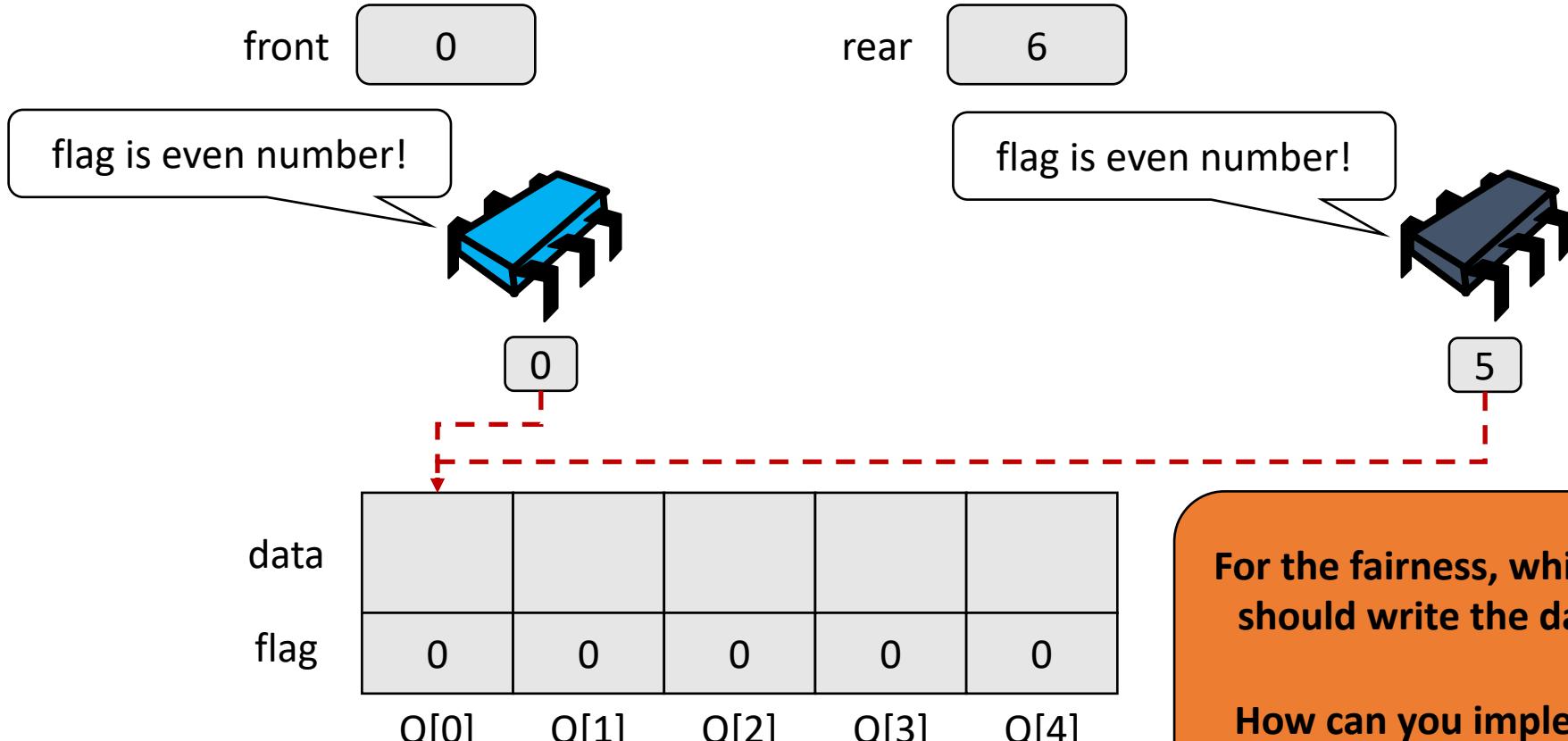
# Enqueue



# Enqueue



# Enqueue



# Evaluation

CPU: 24 x 2 (Hyperthreading enabled)

Number of producer / consumer threads: 16

Number of enqueue / dequeue operations per thread: 1,000,000

```
[jongbin@multicore-24:~/TA/Multicore/lab13$ time ./queue_giantlock
CORRECT!

real    0m5.593s
user    0m4.432s
sys     2m49.276s
```

```
[jongbin@multicore-24:~/TA/Multicore/lab13$ time ./queue_unbounded
CORRECT!

real    0m14.373s
user    7m3.264s
sys     0m8.220s
```

```
[jongbin@multicore-24:~/TA/Multicore/lab13$ time ./queue_bounded
CORRECT!

real    0m1.972s
user    0m56.572s
sys     0m0.848s
```

# Thank You