cis112

Queue

Sait Erkovan
BBBF
Yeditepe University

v2024-11-11

Content

- Motivation
- Case: Printing
- Implementation
- Stack vs Queue
- Why Circular Arrays?
- Applications
  - A simple exercise
  - Stutter in a queue
  - Mirror of a queue
- References

Motivation

Motivation

Queues in computer science [1]

- Operating systems:
  - queue of print jobs to send to the printer
  - queue of programs / processes to be run
  - queue of network data packets to send
- Programming:
  - modeling a line of customers or clients
  - storing a queue of computations to be performed in order
- Real world examples:
  - people on an escalator or waiting in a line
  - cars at a gas station (or on an assembly line)

Case: Printing

Case: Printing

Sending to the printer	Printing	In the order
P1	P1	**P1**

Case: Printing

Sending to the printer	Printing	In the order
P1	P1	P1
P2	P1	P1 P2

Case: Printing

Sending to the printer	Printing	In the order
P1	P1	P1
P2	P1	P1 P2
P3	P1	P1 P2 P3

Case: Printing

Sending to the printer	Printing	In the order
P1	P1	P1
P2	P1	P1 P2
P3	P1	P1 P2 P3
	P2	P1 P2 P3

Case: Printing

Sending to the printer	Printing	In the order
P1	P1	P1
P2	P1	P1 P2
P3	P1	P1 P2 P3
	P2	P1 P2 P3
P4	P2	P1 P2 P3 P4

Case: Printing

Sending to the printer	Printing	In the order
P1	P1	P1
P2	P1	P1 P2
P3	P1	P1 P2 P3
	P2	P1 P2 P3
P4	P2	P1 P2 P3 P4
	P3	P1 P2 P3 P4

Case: Printing

Sending to the printer	Printing	In the order
P1	P1	P1
P2	P1	P1 P2
P3	P1	P1 P2 P3
	P2	P1 P2 P3
P4	P2	P1 P2 P3 P4
	P3	P1 P2 P3 P4
	P4	P1 P2 P3 P4

Case: Printing

Sending to the printer	Printing	In the order
P1	P1	P1
P2	P1	P1 P2
P3	P1	P1 P2 P3
	P2	P1 P2 P3
P4	P2	P1 P2 P3 P4
	P3	P1 P2 P3 P4
	P4	P1 P2 P3 P4
P5	P4	P1 P2 P3 P4 P5

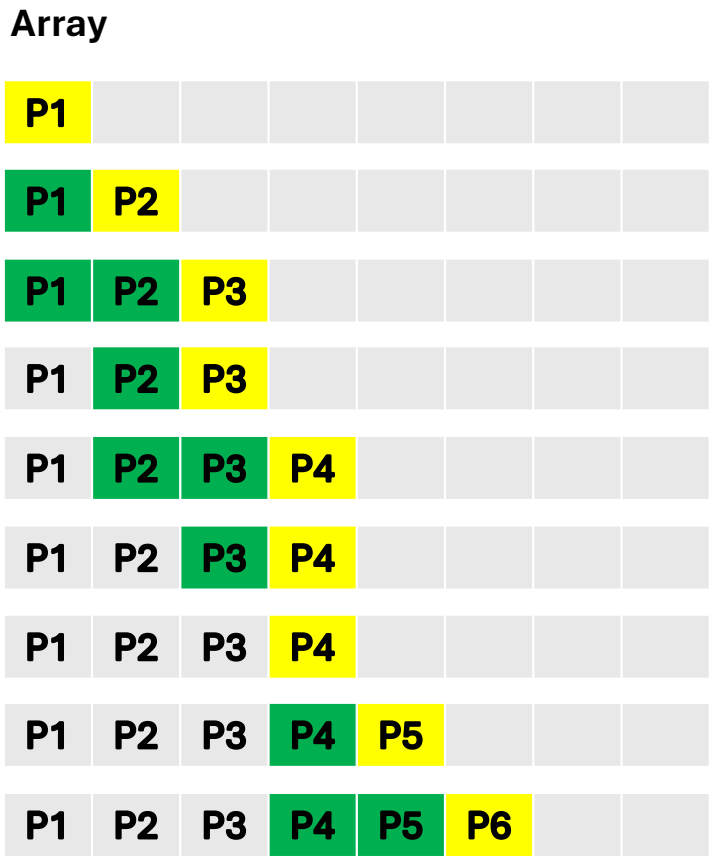
Case: Printing

Sending to the printer	Printing	In the order
P1	P1	P1
P2	P1	P1 P2
P3	P1	P1 P2 P3
	P2	P1 P2 P3
P4	P2	P1 P2 P3 P4
	P3	P1 P2 P3 P4
	P4	P1 P2 P3 P4
P5	P4	P1 P2 P3 P4 P5
P6	P4	P1 P2 P3 P4 P5 P6

Implementation

Observations

- Insertion
  - at location2
- Deletion
  - at location1
- Use array
  - Keep track of location1 and location2

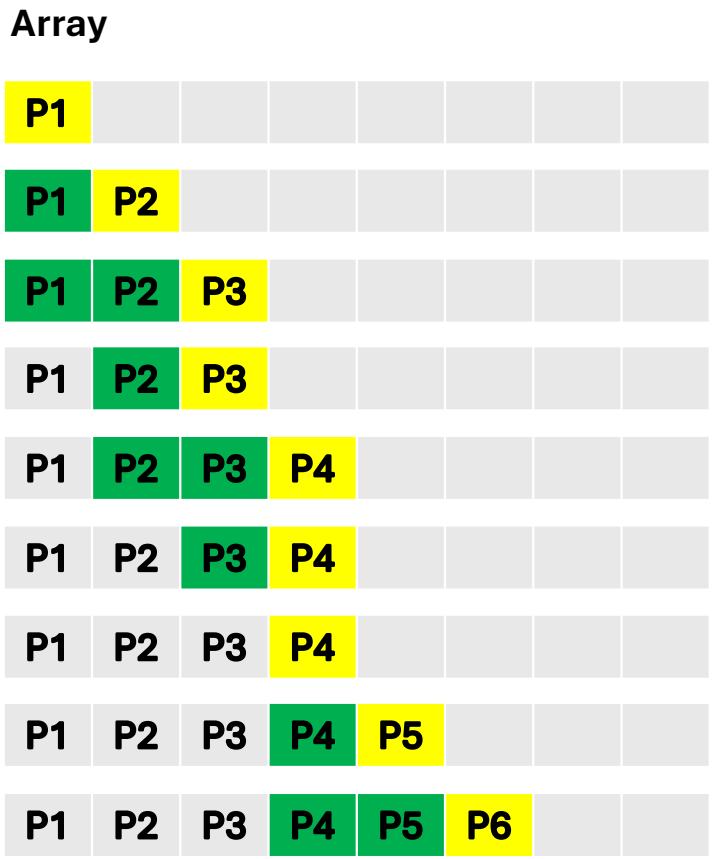


Observations

- Use array
  - Keep track of ~~location1~~ and ~~location2~~ head and tail
- ~~Insertion~~ enqueue(item)
  - at ~~location2~~ tail
- ~~Deletion~~ dequeue()
  - at ~~location1~~ head

```
ENQUEUE(Q, x)
1  Q[Q.tail] = x
2  if Q.tail == Q.size
3        Q.tail = 1
4  else Q.tail = Q.tail + 1
```

```
DEQUEUE(Q)
1  x = Q[Q.head]
2  if Q.head == Q.size
3        Q.head = 1
4  else Q.head = Q.head + 1
5  return x
```



Stack vs Queue

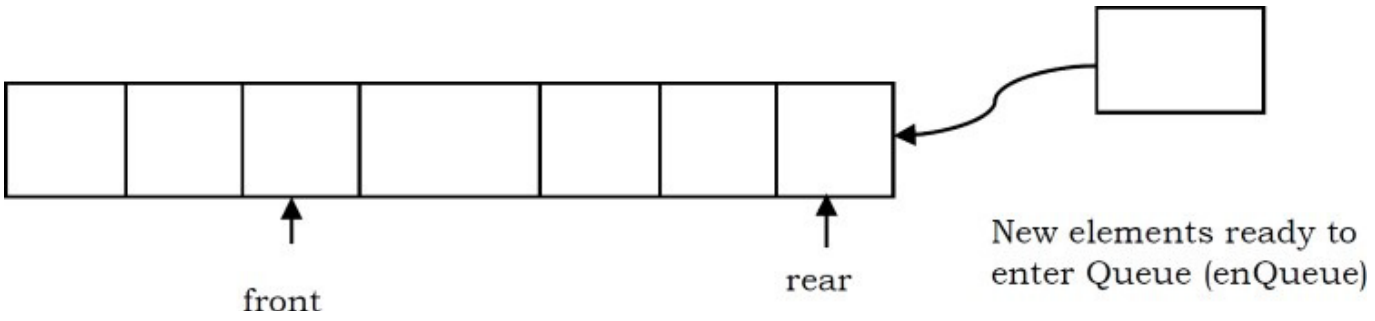
Queue

- A queue is a data structure used for storing data (similar to Stacks)
- In queue, the order in which data arrives is important.
- A *queue* is an ordered list in which insertions are done at one end (*tail*) and deletions are done at other end (*head*).
- The first element to be inserted is the first one to be deleted.
- Hence, it is called **First in First out (FIFO)**
  - (it is LIFO in Stack)

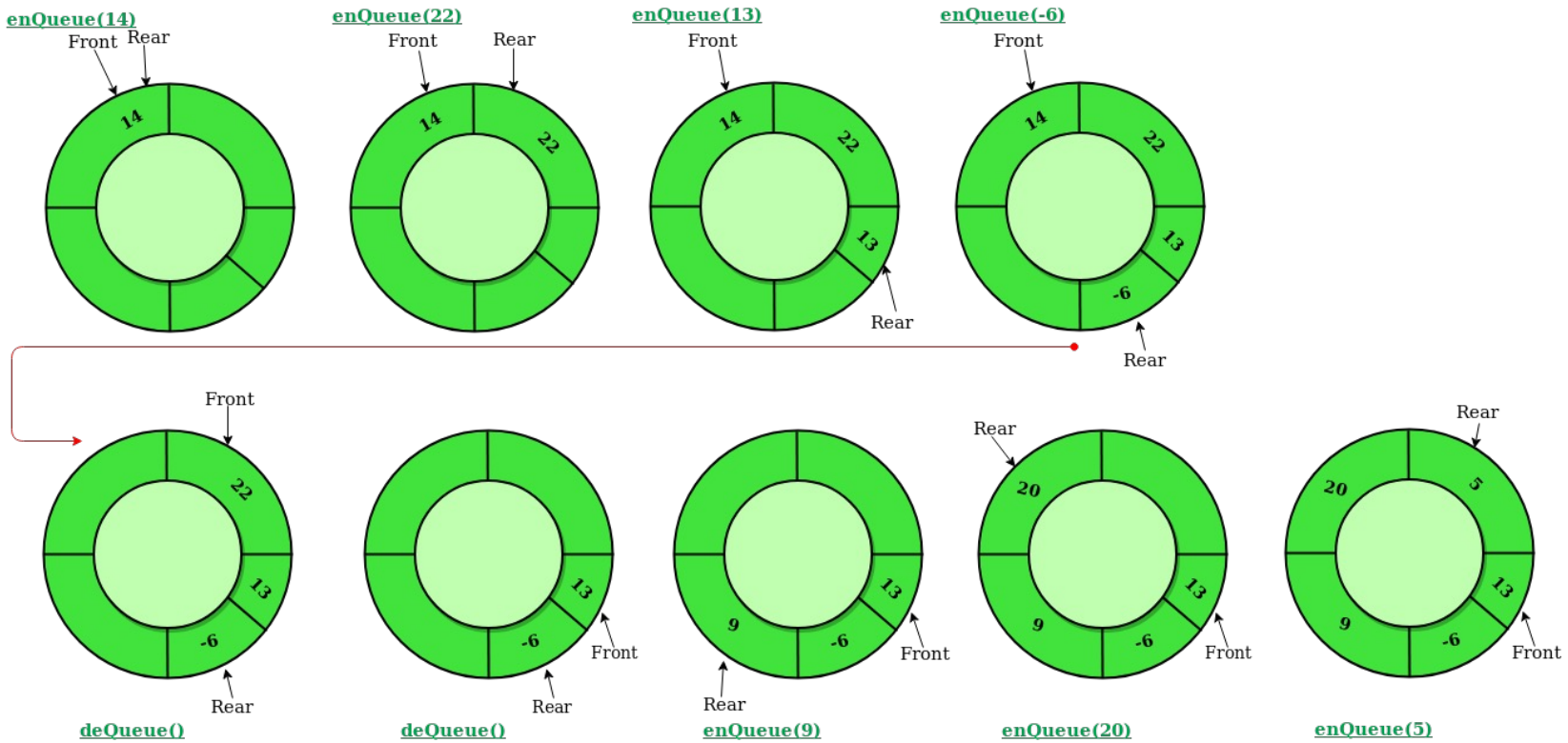
Why Circular Arrays?

Why Circular Arrays?

- In the example shown below, it can be seen clearly that **the initial slots of the array are getting wasted**.
- So, simple array implementation for queue is not efficient.
- To solve this problem we assume the arrays as **circular arrays**.



Example


enQueue(14)
Front
Rear
14
enQueue(22)
Front
Rear
14
22
enQueue(13)
Front
14
22
13
Rear
enQueue(-6)
Front
14
22
13
-6
Rear
Front
22
13
-6
Rear
deQueue()
13
-6
Front
Rear
deQueue()
13
9
-6
Front
Rear
enQueue(9)
Rear
20
13
9
-6
Front
enQueue(20)
Rear
20
5
13
9
-6
Front
enQueue(5)

Applications

A simple exercise

- Illustrate the result of each operation in the sequence on an initially empty queue Q stored in array Q[1:6].
  - ENQUEUE(Q,4),
  - ENQUEUE(Q,1),
  - ENQUEUE(Q,3),
  - DEQUEUE(Q),
  - ENQUEUE(Q,8),
  - DEQUEUE(Q)

Stutter in a queue

- Write a method `stutter` that accepts a queue of integers as a parameter and replaces every element of the queue with two copies of that element.
  - `front [1, 2, 3] back`
    becomes
    `front [1, 1, 2, 2, 3, 3] back`

Mirror of a queue

- Write a method `mirror` that accepts a queue of strings as a parameter and appends the queue's contents to itself in reverse order.
  - `front [a, b, c] back`
    becomes
    `front [a, b, c, c, b, a] back`

References

- [1] S. Reges, and M. Stepp, Building Java Programs: A Back to Basics Approach, 5th edition.

- [2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms. MIT Press, 2022.