DATA STRUCTURES

By

Dr. Yasser Abdelhamid

الجامعة المصرية للتعلم الإلكتروني الأهلية
EELU
THE EGYPTIAN E-LEARNING UNIVERSITY

RESOURCES

- http://javatpoint.com/

OUTLINE

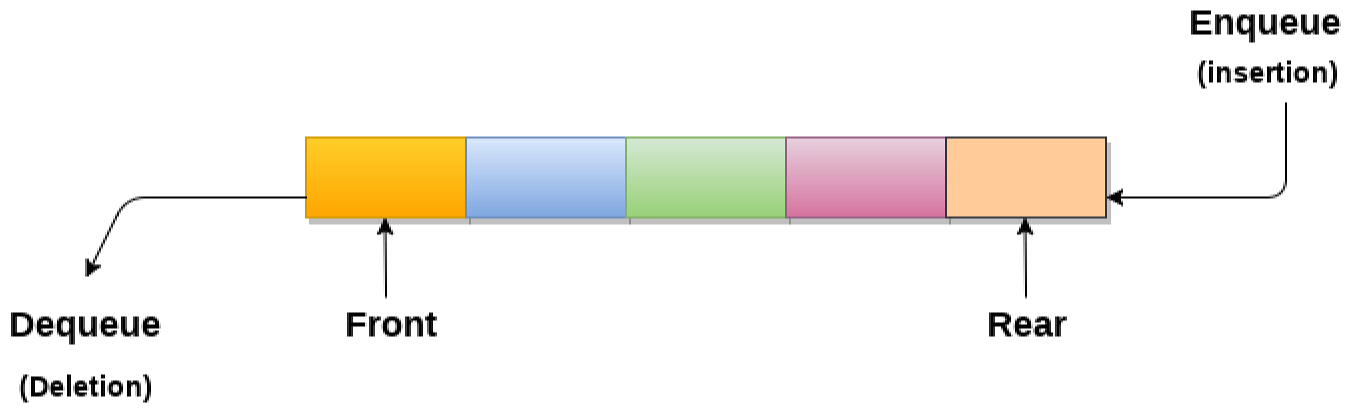
- Queues

QUEUES

- Queue is a data structure in which the elements are added at one end, called the **rear**, and deleted from the other end, called the **front**.
- The elements in the middle between front and rear are not accessible.
- A queue is a First In First Out (FIFO) data structure.

QUEUES

- **Examples:**
  - Tiller waiting line.
  - Elevator waiting line.
  - Print job waiting list



QUEUES OPERATIONS

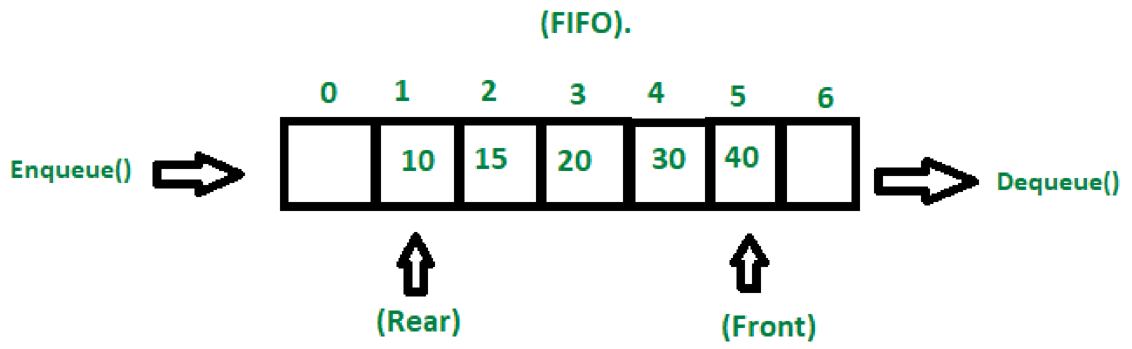
- **Main operations:**
  - enqueue
  - dequeue
- **Auxiliary operations:**
  - initializeQueue
  - isEmptyQueue
  - isFullQueue
  - front
  - rear

QUEUE EXCEPTIONS

- Adding an element to a full queue and removing an element from an empty queue would generate errors or exceptions
- Queue overflow exception
- Queue underflow exception

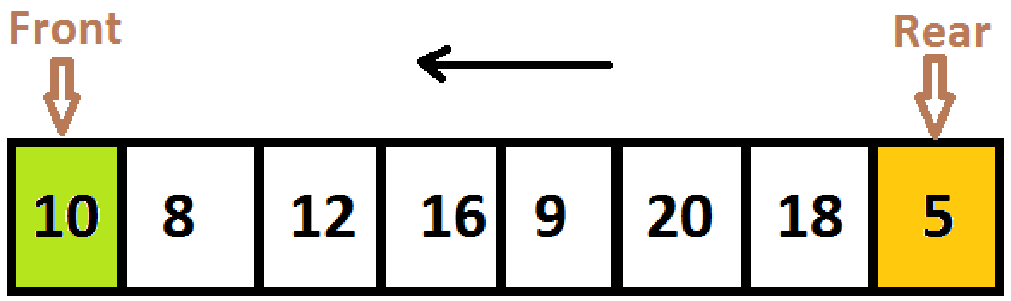
IMPLEMENTATION OF QUEUES USING ARRAYS

- An array to store the queue elements
- Front: points at the first element of the queue.
- Rear: points at the last element of the queue.
- maxQueueSize: specifies the maximum size of the queue.



EXAMPLE

Queue Data Structure (First In First Out)

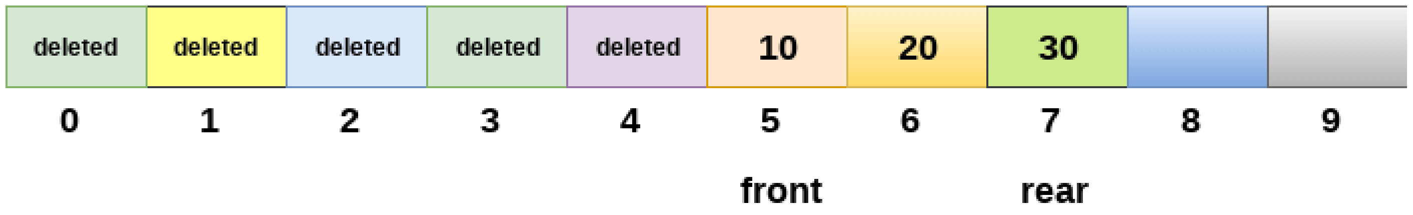


Enqueue(10)
Enqueue(8)
Enqueue(12)
Enqueue(16)
Enqueue(9)
Enqueue(20)
Enqueue(18)
Dequeue() -->10
Dequeue() -->8
Dequeue() -->12
Dequeue() -->16
Dequeue() -->9
Dequeue() -->20
Dequeue() -->18

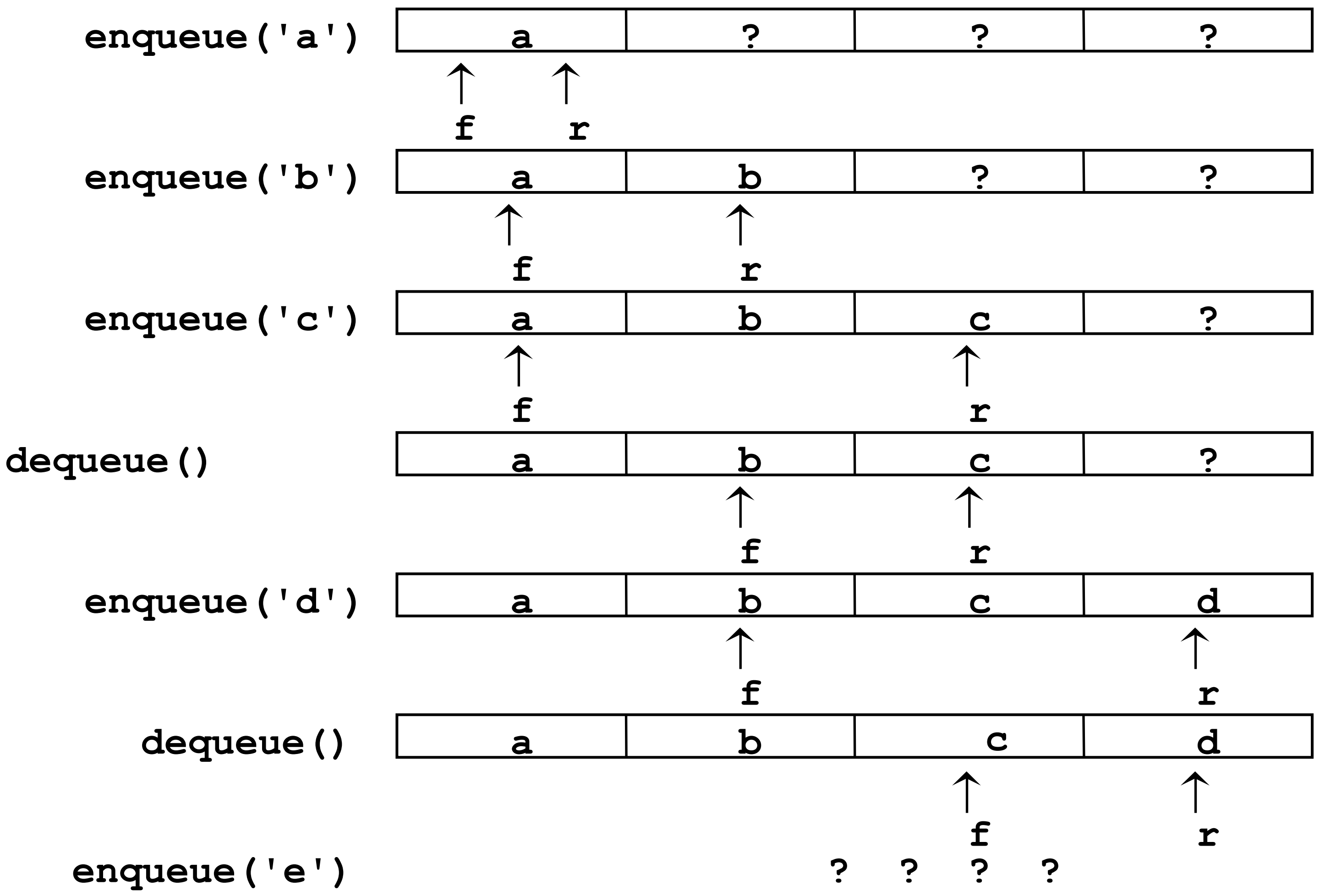
PROBLEMS OF ARRAY IMPLEMENTATION

- Arrays have fixed sizes.
- After a number of insertion and deletion operations, Rear will point at the last array position.

- Solutions
  - Slide all of the queue elements toward the first array position.
  - Use a circular array

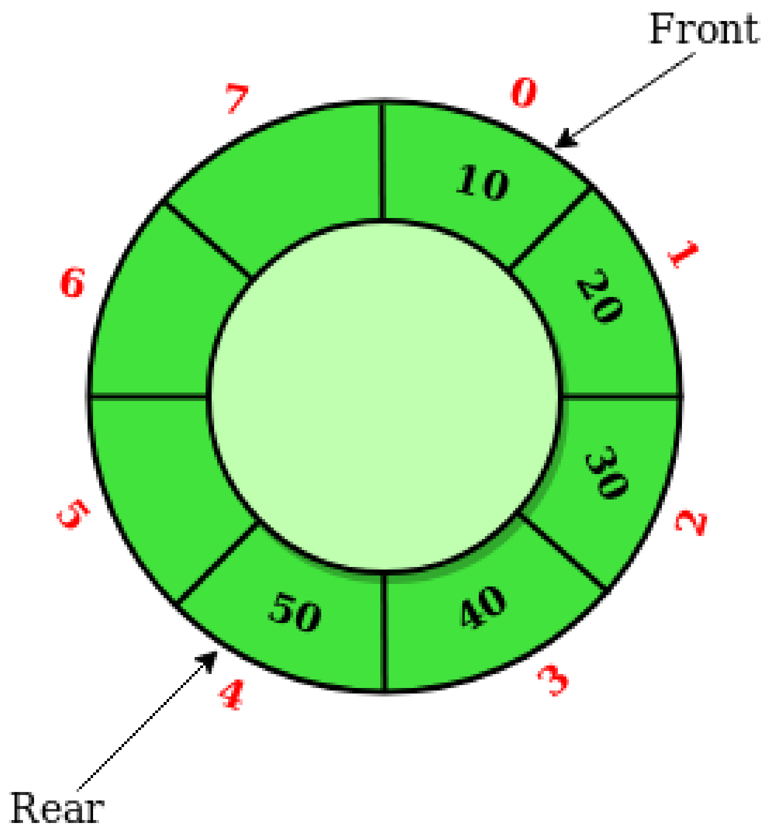


HOW IT WORKS



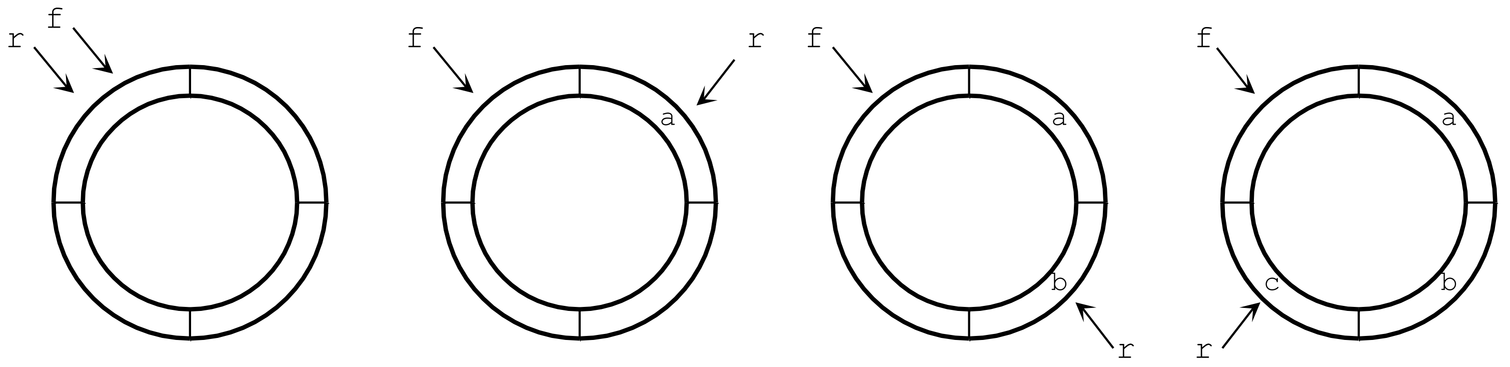
enqueue('a')
a ? ? ?
f r
enqueue('b')
a b ? ?
f r
enqueue('c')
a b c ?
f r
dequeue()
a b c ?
f r
enqueue('d')
a b c d
f r
dequeue()
a b c d
f r
enqueue('e')
? ? ? ?

CIRCULAR QUEUES



next_index= (current_index + 1) % MaxQueueSize

HOW CIRCULAR QUEUES WORK



Empty

Full

QUEUE IMPLEMENTATION

```
public class CircularQueue {
    private int[] queue;
    private int front;
    private int rear;
    private int size;

    public CircularQueue(int k) {
        queue = new int[k];
        front = -1;
        rear = -1;
        size = 0;
    }
```

QUEUE IMPLEMENTATION

```
public boolean isEmpty() {
        return size == 0;
    }

    public boolean isFull() {
        return size == queue.length;
    }

    public int size() {
        return size;
    }
```

QUEUE IMPLEMENTATION

```
public void enqueue(int x) {
      if (isFull()) {
         System.out.println("Queue is full.");
         return;
      }
      if (isEmpty()) {
         front = 0;
         rear = 0;
      } else {
         rear = (rear + 1) % queue.length;
      }
      queue[rear] = x;
      size++;
   }
```

QUEUE IMPLEMENTATION

```
public int dequeue() {
        if (isEmpty()) {
            System.out.println("Queue is empty.");
            return -1;
        }
        int x = queue[front];
        if (front == rear) {
            front = -1;
            rear = -1;
        } else {
            front = (front + 1) % queue.length;
        }
        size--;
        return x;
    }
```

QUEUE IMPLEMENTATION

```
public int peek() {
        if (isEmpty()) {
            System.out.println("Queue is
empty.");
            return -1;
        }
        return queue[front];
    }
```

الجامعة المصرية
للتعلم الإلكتروني الأهلية

EELU
THE EGYPTIAN E-LEARNING UNIVERSITY

THANK YOU