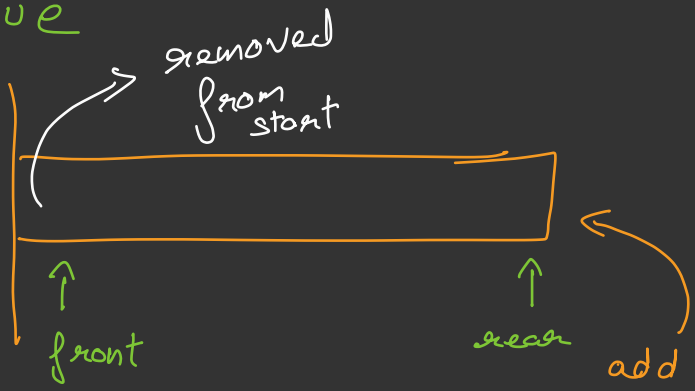
Queue




```
add (1)  // Enqueue
add (2)
add (3)

remove() // dequeue
peek()  // get value of front element
```

Note

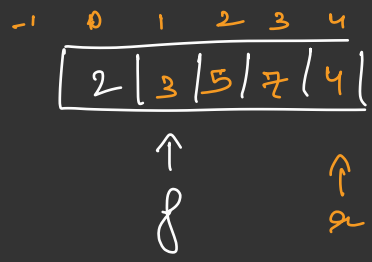
FIFO → First In First Out

Implementation

(1) Arrays

```
Class Queue {
    int data[];
    int size;
    int front;
    int rear;
```

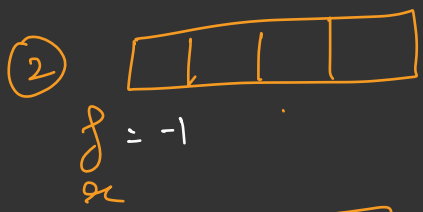
add
remove
peek
isEmpty
size



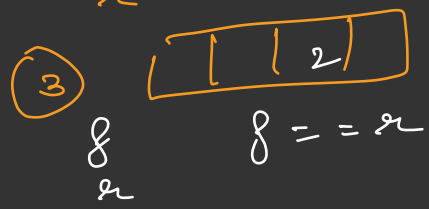
add(2)
add(3)
add(5)
add(7)
remove()
add(4)
add(8)

S = 3

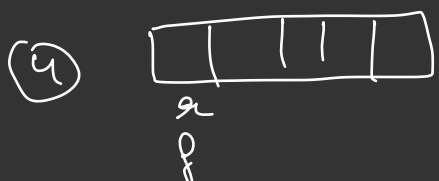
Remove

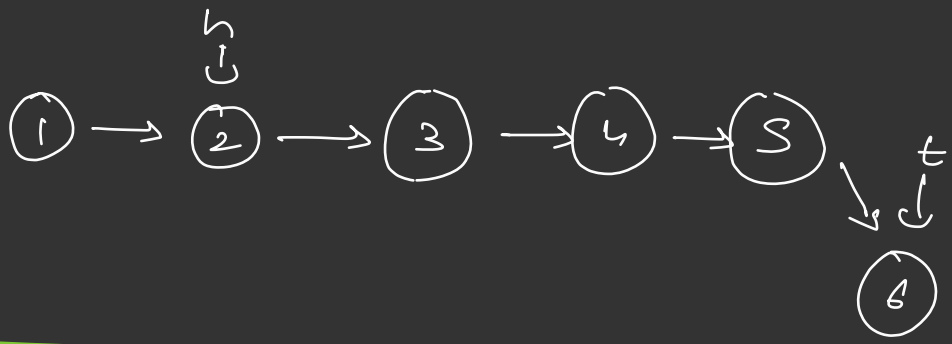


No element
Queue Underflow



size = 0
f = -1
r = -1





```
Stack<Integer> st = new Stack<>()
```

```
Queue<Integer> q = new Queue<>();
```

Queue → Interface

new Queue<>(); → ✗

① LinkedList

② ArrayDeque

Ques · 1, 3, 10, 2, 9, 11, 4, 6

k = 4

⇒

6, 4, 11, 9, 1, 3, 10, 2,

reversed

① Enqueue all

② n - k

q.add(q.remove())

9, 11, 4, 6, 1, 3, 10, 2

③ st.push(q.remove) k

1, 3, 10, 2

q.add(st.pop()) → R

1, 3, 10, 2, 6, 4, 11, 9

Repeat (2) step

6, 4, 11, 9, 1, 3, 10, 2