B.Sc. in (Hons) Computing (Information Systems

ICT 2306 – Data Structures and Algorithms
FIFO Queue

8:07 AM 87%

Vivek
online

Willst du zu Abend essen? 9:04 PM
Willst du zu Abend essen? 9:04 PM
Willst du zu Abend essen? 9:07 PM
Willst du zu Abend essen? 9:08 PM
Willst du zu Abend essen? 9:32 PM
Willst du zu Abend essen? 9:32 PM
Willst du zu Abend essen? 9:52 PM
Great 10:46 PM

TODAY

Hey vivek, Here's wishing you a great birthday. Hope you have many many wonderful years ahead 7:47 AM
Lass uns nach der Einreichung zum Mittagessen aufholen 8:06 AM
Woher kommst du? 8:07 AM

Type a message

LNBTI
LANKA NIPPON BIZTECH INSTITUTE

ACKNOWLEDGEMENT

Presented by:

Mr. Janith Perera
Ms. Hansika Samanthi
Ms. Ridmi Wimalasiri

Contributors:

Mr. Janith Perera
Ms. Nimali Wasana
Ms. Hansika Samanthi
Ms. Ridmi Wimalasiri

Reviewed by: Mr. Janith Perera

RESOURCES

- Course page of the course
  - https://lms.lnbti.lk/course/view.php?id=692
- References available for the course.
  - M. Weiss. Data Structures and Problem Solving using Java. 4th Edition. Addison-Wesley, 2006.
  - Joel Grus, Data Science from Scratch: First Principles with Python, 1st ed. O'Reilly, 2015.
  - Cathy O’Neil and Rachel Schutt. Doing Data Science, Straight Talk from The Frontline. O’Reilly. 2014.
  - Foster Provost and Tom Fawcett. Data Science for Business: What You Need to Know about Data Mining and Data-analytic Thinking, 1st ed. O'Reilly, 2013.
  - Kevin P. Murphy. Machine Learning: A Probabilistic Perspective, 1st ed. The MIT Press, 2012.
- Online resources for the section

  https://docs.oracle.com/en/java/javase/22/docs/api/java.base/java/util/packagesummary.html#JavaCollectionsFramework

AGENDA

1. FIFO Queue
2. Implementations of FIFO Queue
3. Variations of FIFO Queue

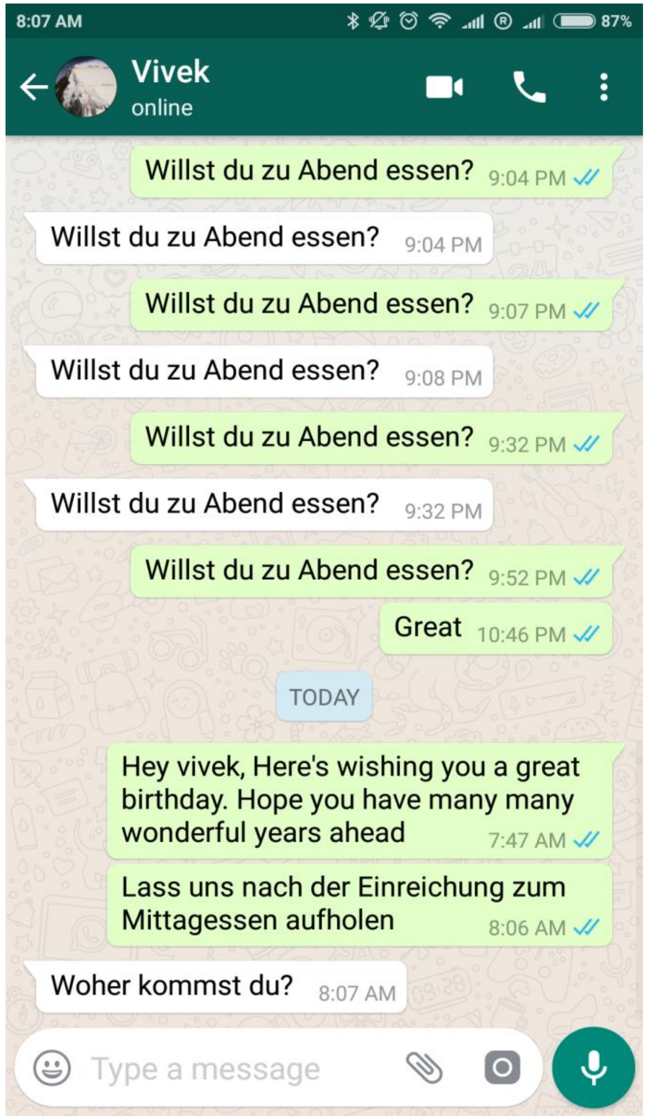
INTENDED LEARNING OUTCOMES (ILO)

By the end of this section, students should be able to:

- ILO1: Describe what is a FIFO queue and its basic operations.
- ILO2: Explain the applications of FIFO queue.
- ILO3: Implement FIFO queues using different data structures.
- ILO4: Discuss about the variations of the FIFO queues.

REAL LIFE EXAMPLES

ATM Queue


8:07 AM 87%

Vivek
online

Willst du zu Abend essen? 9:04 PM

Willst du zu Abend essen? 9:04 PM

Willst du zu Abend essen? 9:07 PM

Willst du zu Abend essen? 9:08 PM

Willst du zu Abend essen? 9:32 PM

Willst du zu Abend essen? 9:32 PM

Willst du zu Abend essen? 9:52 PM

Great 10:46 PM

TODAY

Hey vivek, Here's wishing you a great birthday. Hope you have many many wonderful years ahead 7:47 AM

Lass uns nach der Einreichung zum Mittagessen aufholen 8:06 AM

Woher kommst du? 8:07 AM

Type a message

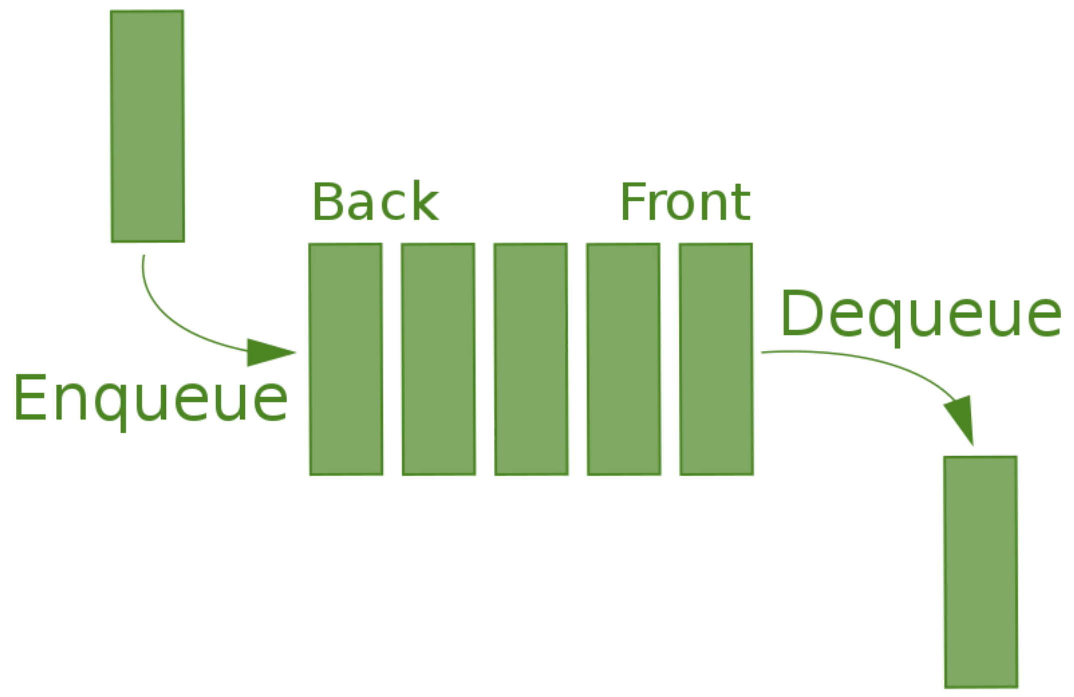
Message Queue

Vehicle Queue

1. FIFO Queue

FIFO QUEUE

- FIFO Queue follows the First In First Out (FIFO) rule.
- Element at the front is the next processing one(head).



BASIC OPERATIONS IN FIFO QUEUE

• Enqueue	: Add an element to the rear end(back) of the FIFO queue.
• Dequeue	: Remove an element from the front of the FIFO queue.
• IsEmpty	: Check if the FIFO queue is empty.
• IsFull	: Check if the FIFO queue is full. (If it is bounded)
• Peek	: Get the value of the front of the FIFO queue without removing it.

APPLICATIONS OF FIFO QUEUES

- There are various queues quietly doing their job in our computer's (or the network's) operating system.
- Printer queue where print jobs wait for the printer to be available.
- A queue also stores keystroke data as we type at the keyboard.
  - This way, if we are using a word processor but the computer is briefly doing something else when we hit a key, the keystroke won't be lost; it waits in the queue until the word processor has time to read it.
  - Using a queue guarantees the keystrokes stay in order until they can be processed.

2. Implementation of FIFO Queue

IMPLEMENTATION OF FIFO QUEUE

- FIFO queue operates in First-In-First-Out manner.
- FIFO queue needs to know who is at the head, because head is the next processing one.
- The FIFO queue is opened from the both ends.

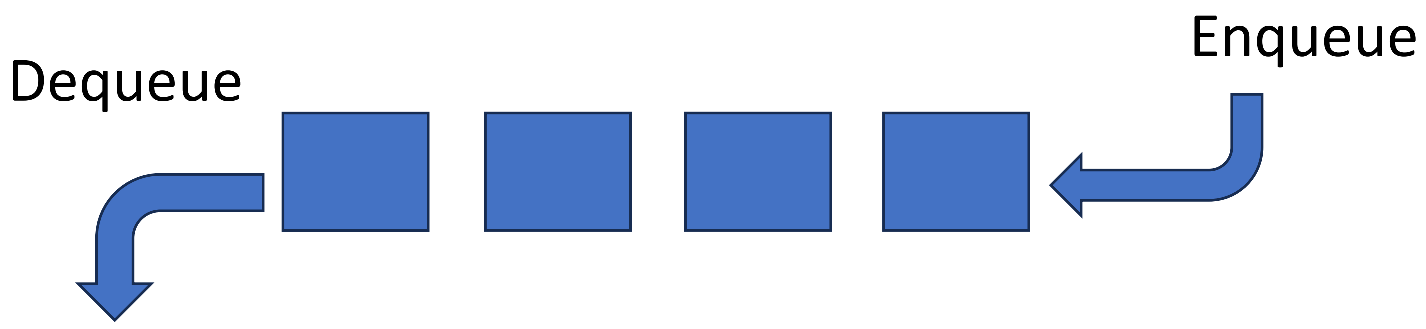
IMPLEMENTATION OF FIFO QUEUE CONTD…

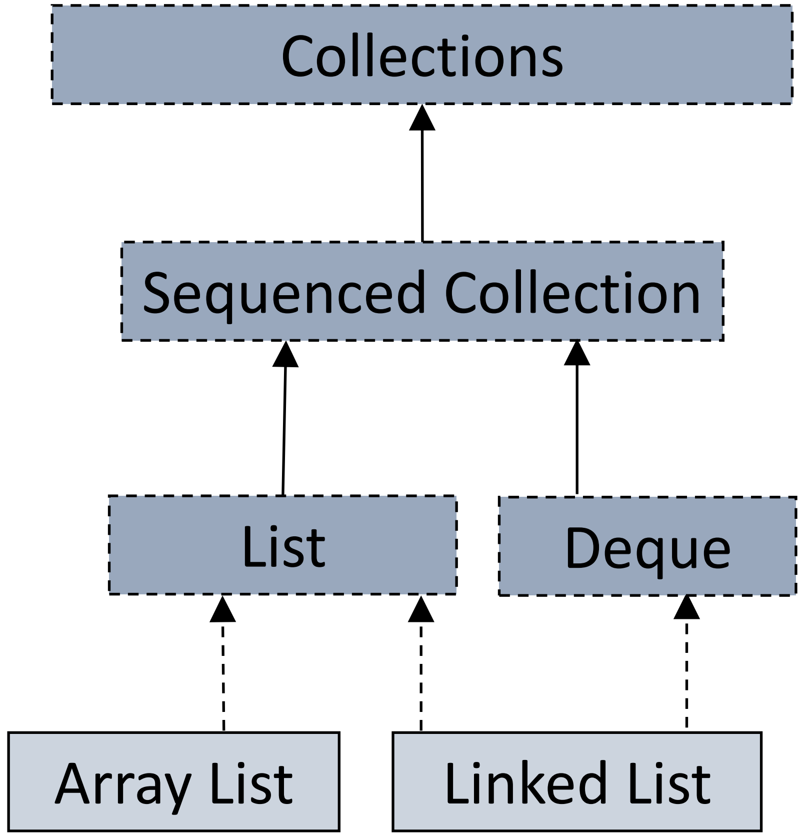
- There are many ways of implementing a FIFO queue.
  - Using a sequenced collection (Like ArrayList – use removeFirst(),addLast(E e))
  - Using a deque (Like ArrayDeque – use remove(), add(E e))
  - Implement a dequeue-friendly FIFO Queue Using LIFO Queues (stacks)
  - Implement an enqueue-friendly FIFO Queue Using LIFO Queues (stacks)
  - Using an array (from scratch)
  - Etc.

FIFO QUEUE IMPLEMENTATION

Using a Sequenced Collection

- List has methods to add elements at the end (rear), and to remove elements from the start (front).
- Also, it can return the head element of the queue.

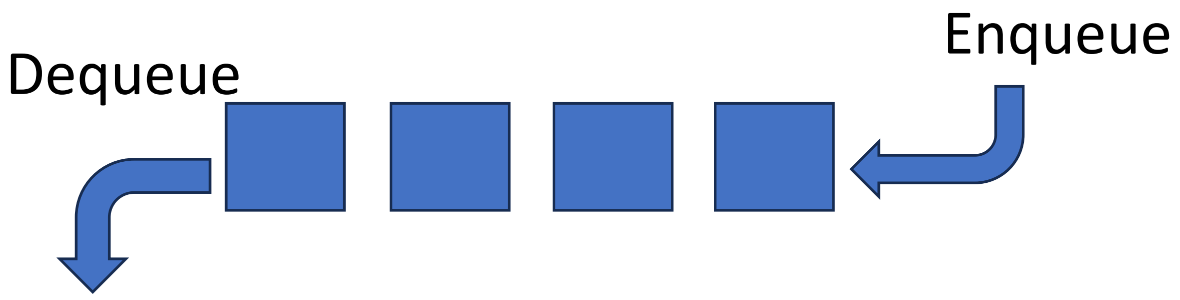


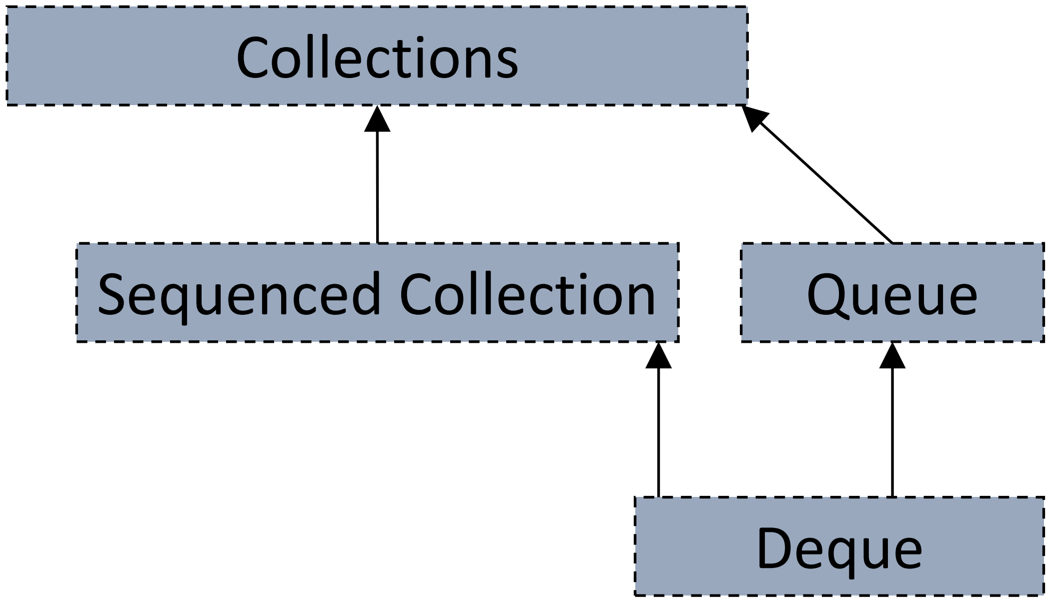


FIFO QUEUE IMPLEMENTATION

Using a Deque

- A deque can be used as a queue.
- Elements are added at the end of the deque and removed from the beginning.





FIFO QUEUE IMPLEMENTATION

Using Two Stacks

- A queue can be implemented using two stacks.
- There are two methods:
  - Method 1 - Dequeue-friendly FIFO Queue Using LIFO Queues
  - Method 2 - Enqueue-friendly FIFO Queue Using LIFO Queues

METHOD 1

Dequeue-friendly FIFO Queue Using LIFO Queues

- In this approach, enqueue operations are costly, while dequeue operations are efficient.
- Also called as enqueue operations costly.

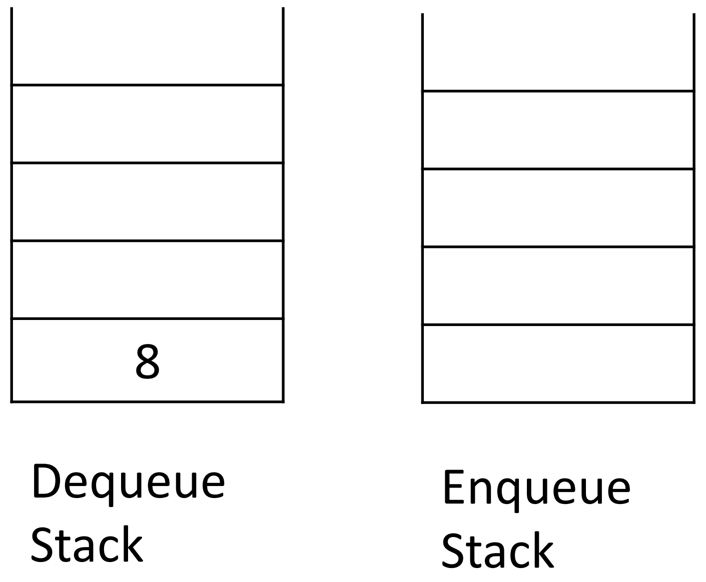
Steps for Enqueue operation

1. while the enqueue stack is empty, push the element to stack dequeue stack.
2. else, push all the elements from enqueue stack to dequeue stack, and push the element to the top of the enqueue stack.
3. Now, Push all the elements from dequeue stack to enqueue stack.

ENQUEUE OPERATION

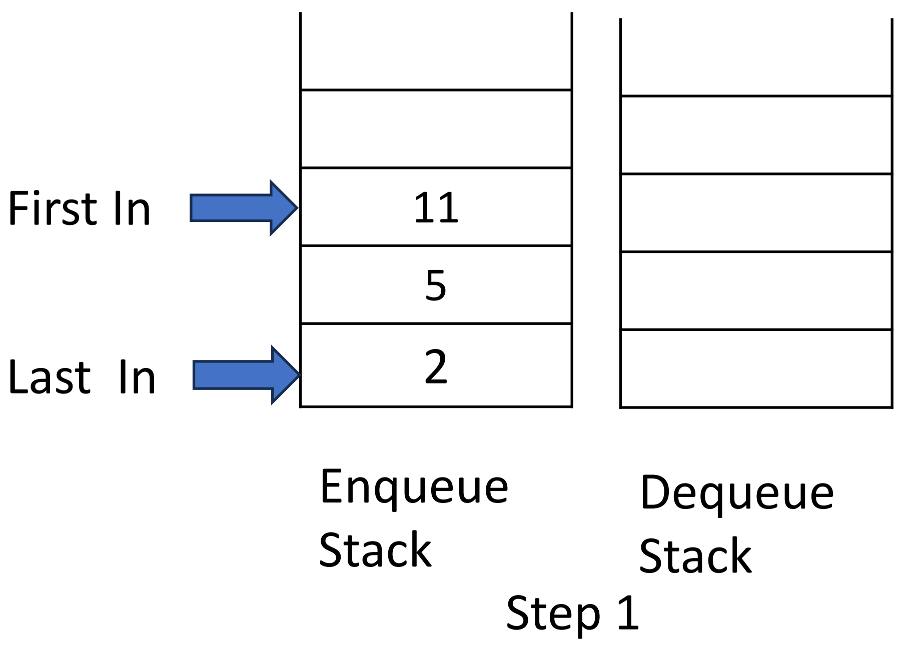
Step 1 : When Dequeue stack is empty.

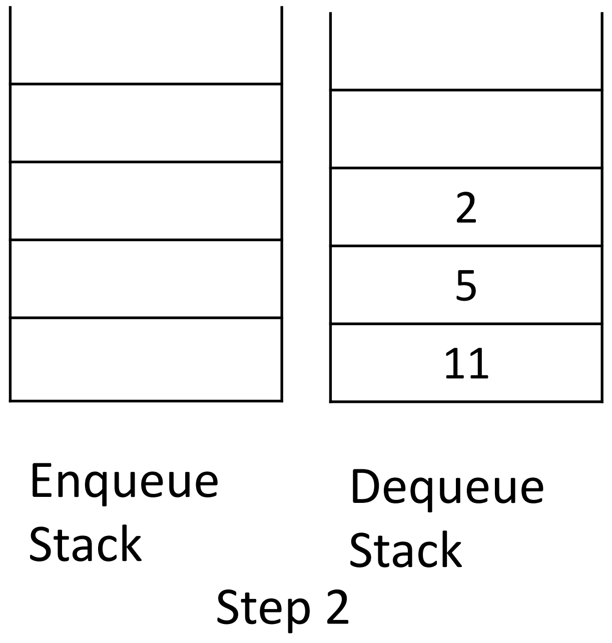
Push the new element to Dequeue stack.



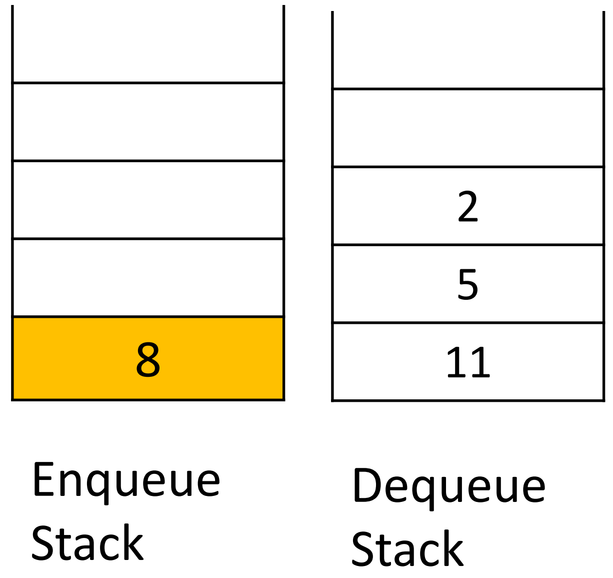
ENQUEUE OPERATION CONTD…

Step 1 : When enqueue stack is not empty.

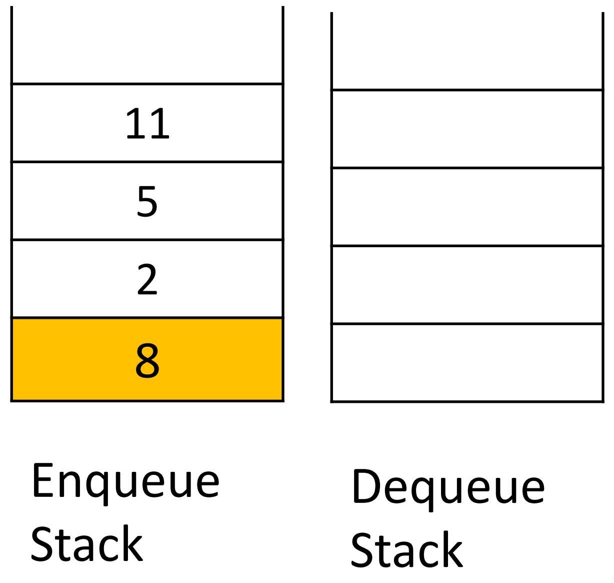




ENQUEUE OPERATION CONTD…



Step 3: New element is 8



Step 4

DEQUEUE OPERATION

- If the stack is empty, return -1.
- Otherwise, pop and return the top element of the dequeue stack.

METHOD 2

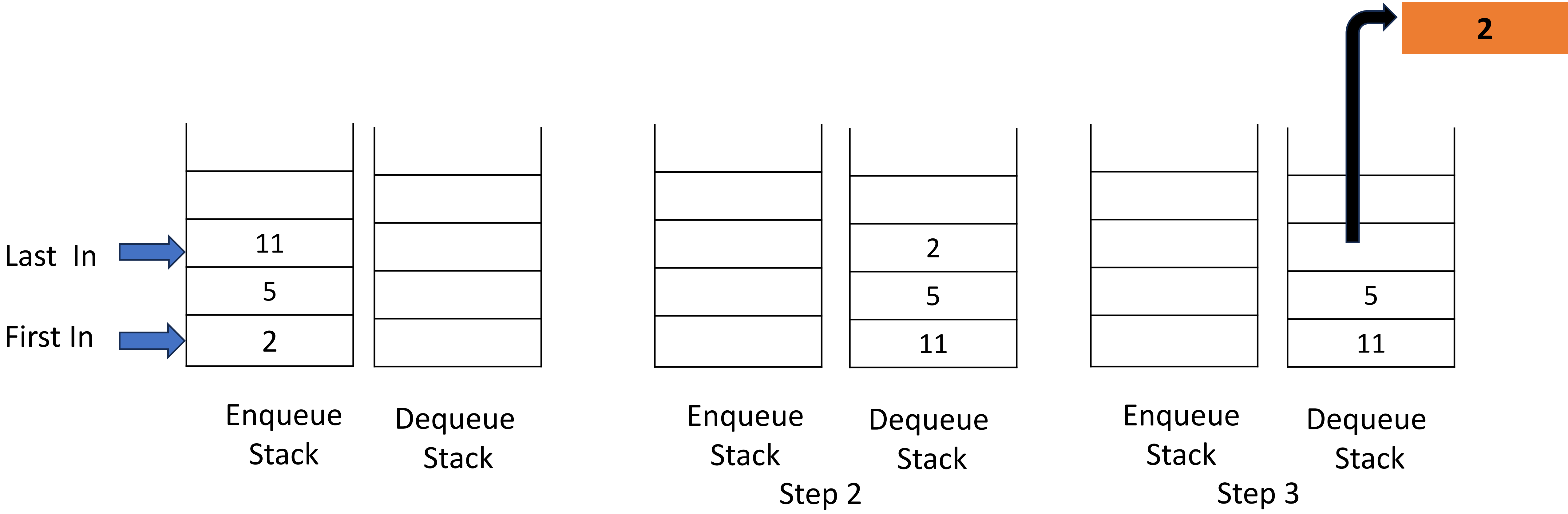
Enqueue-friendly FIFO Queue Using LIFO Queues

- In this approach, dequeue operations are costly, while enqueue operations are efficient.
- Also called as dequeue operations costly.
- Steps for Enqueue operation.
    - When enqueuing an element, simply push the new element onto enqueue stack.

DEQUEUE OPERATION

1. If enqueue stack and dequeue stack are empty, return -1 as there is no element to remove.
2. If dequeue is empty, push all the elements of stack from enqueue stack to dequeue stack.
3. Remove the top element of the dequeue.

DEQUEUE OPERATION CONTD…



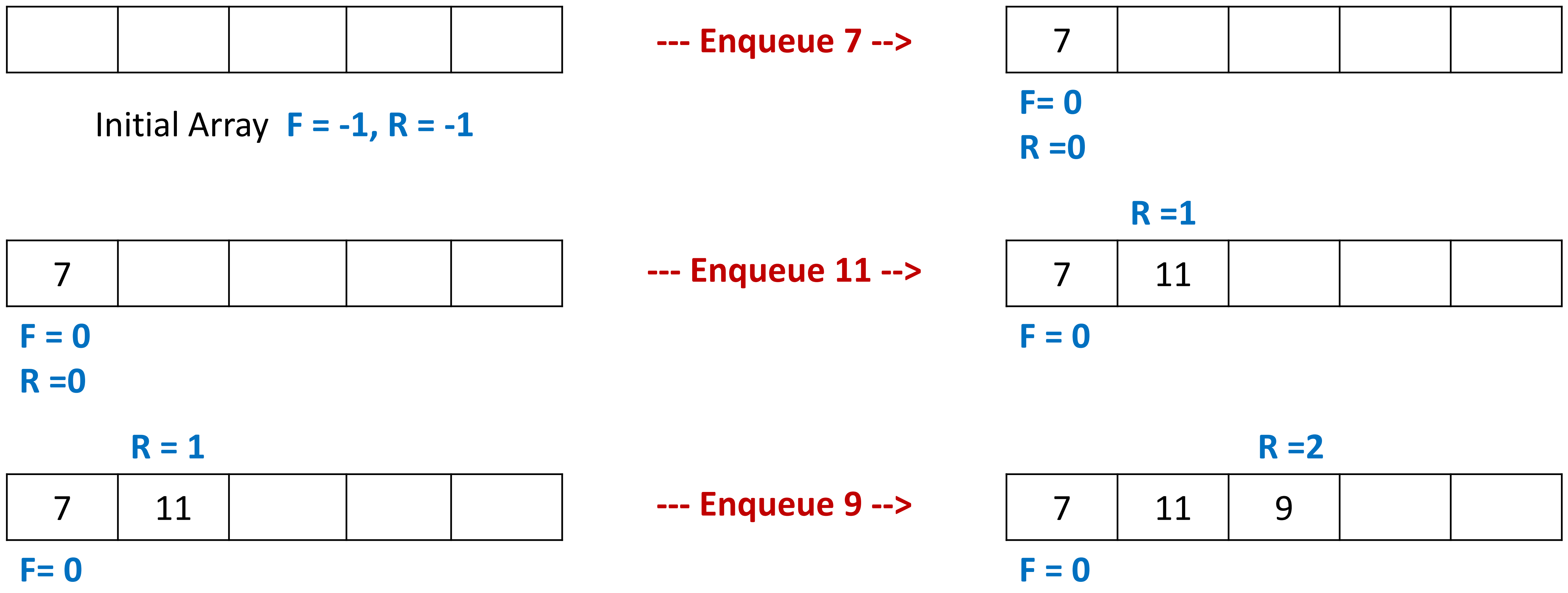
FIFO QUEUE IMPLEMENTATION

Using an Array

- In array implementation there should be a front and rear.
- Both rear and front needs to be -1 initially.

- Enqueue Operation:
- Increment the rear pointer by one and insert the new element at the index indicated by the rear pointer.
- If the queue is empty, set both front and rear pointers to 0 after enqueuing the first element.
- If the rear pointer reaches the size of the array minus one, the enqueue operation cannot be performed.

USING AN ARRAY

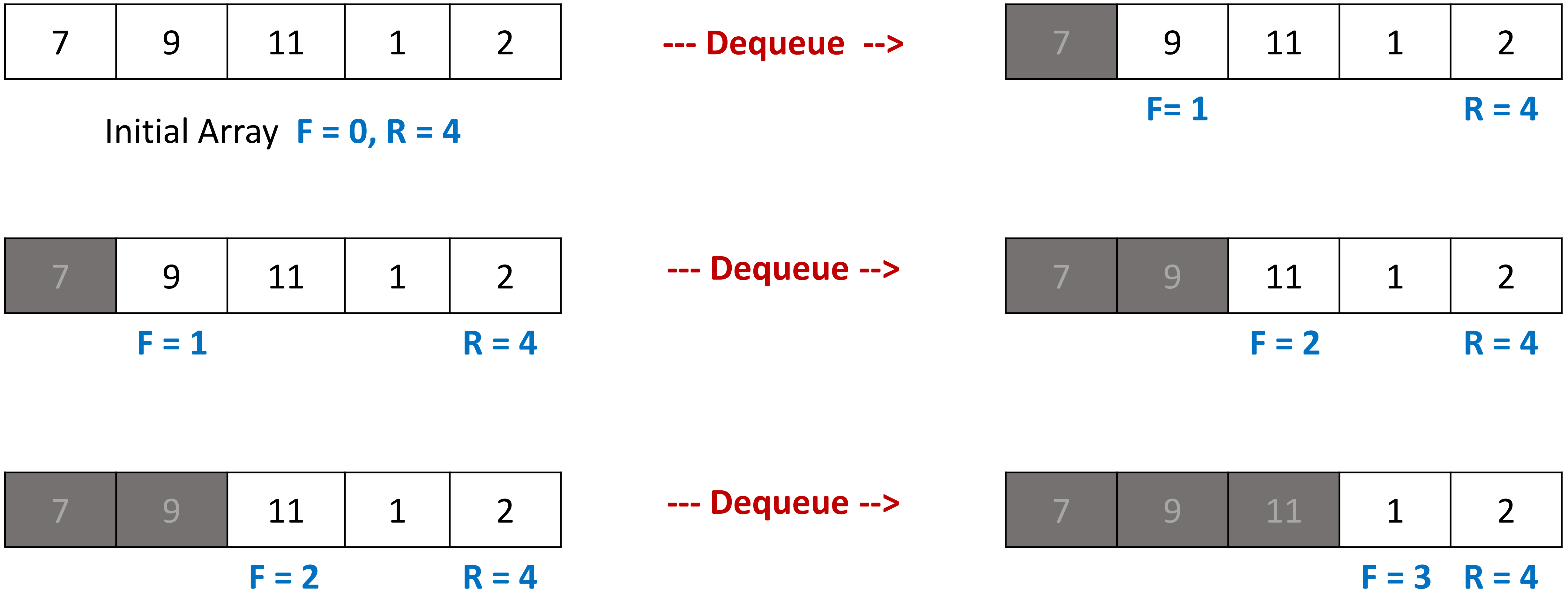


R – Rear F - Front

DEQUEUE OPERATION

- Retrieve the element at the index indicated by the front pointer, and then increment the front pointer by one.
- If the queue becomes empty after dequeuing, reset both front and rear pointers to -1 to indicate an empty queue.
- Ensure that the queue is not empty before performing the dequeue operation.

DEQUEUE OPERATION



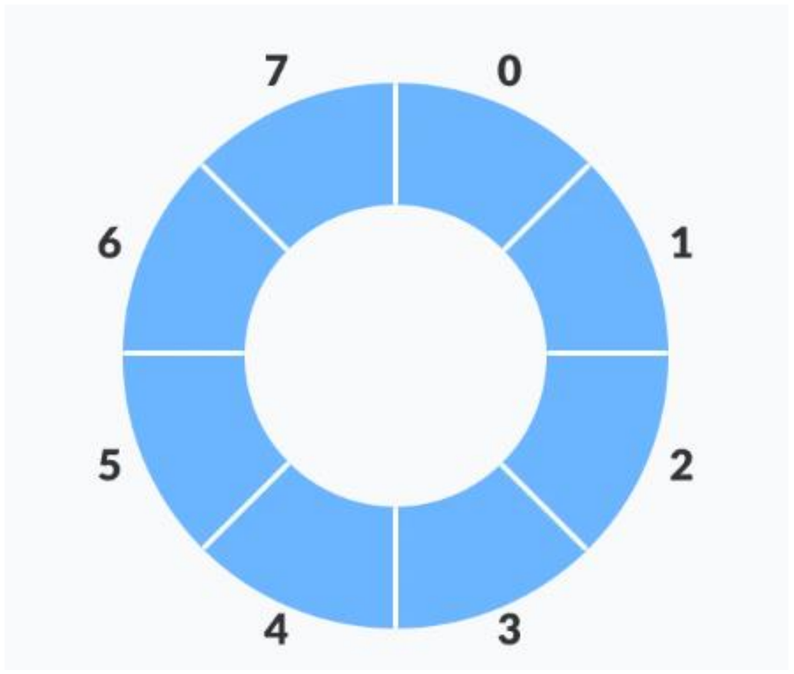
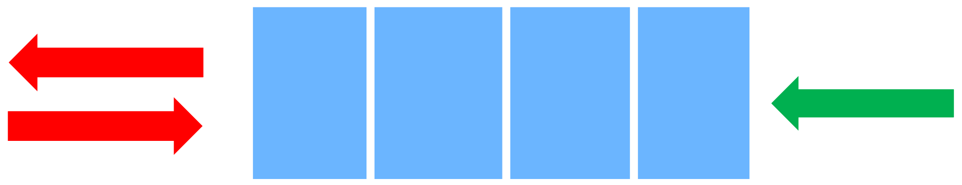
R – Rear F - Front

LIMITATIONS OF FIFO QUEUES

- Linear Structure:
  - FIFO queues have a linear structure, meaning that once the queue is full, it cannot accept any more elements even if there are empty slots available after dequeuing some elements.
- Space Efficiency:
  - There might be wasted space at the beginning of the array where elements have been dequeued.
- Flexibility:
  - FIFO queues have a strict first-in-first-out policy, which might not be suitable for all scenarios.
  - Once an element is added mistakenly, it cannot be removed until it is the next processing element.
  - Etc

3. Variations of FIFO Queues

VARIATIONS OF FIFO QUEUES

Circular Queue	Steque
	

CIRCULAR QUEUE

- It is similar to a simple queue, but the last element is connected to the first element, creating a circular structure.
- This allows for efficient use of memory.
- The circular queue solves the major limitation of the normal queue.
- In a normal queue, after a bit of insertion and deletion, there will be non-usable empty space.

CIRCULAR QUEUE CONTD…

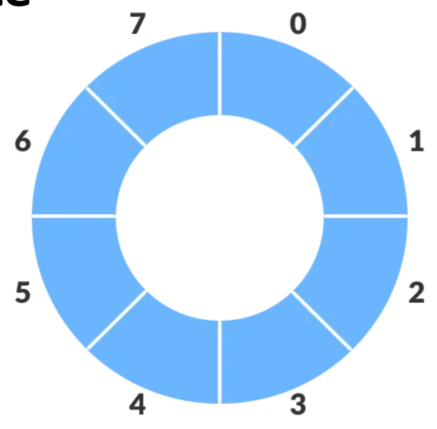
- To avoid the problem of not being able to insert more items into the queue even when it's not full, the Front and Rear arrows *wrap around* to the beginning of the array.
- The result is a *circular queue*

OPERATIONS OF A CIRCULAR QUEUE

- enQueue(value)
    - Used to insert a new value in the Circular Queue. This operation takes place from the end of the Queue.
- deQueue()
    - Used to delete a value from the Circular Queue. This operation takes place from the front of the Queue.
- Peek()
    - Return the value in the queue without removing it.
- IsEmpty()
- IsFull()

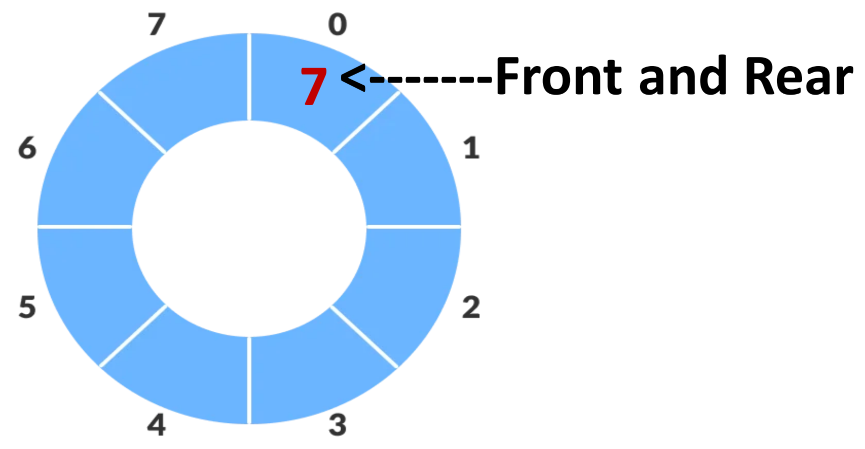
OPERATIONS OF A CIRCULAR QUEUE

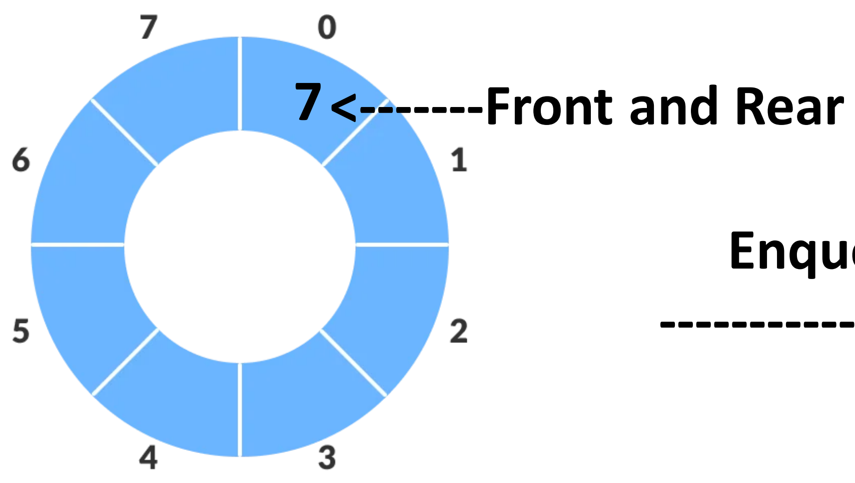
Initial State



Enqueue 7

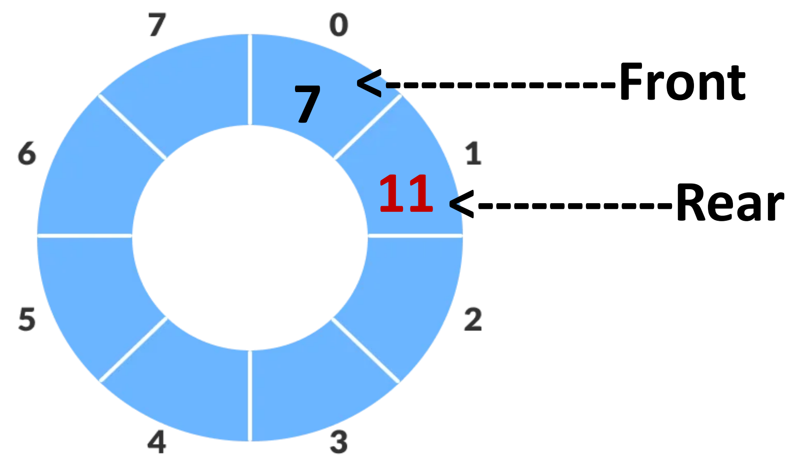
----------------------->





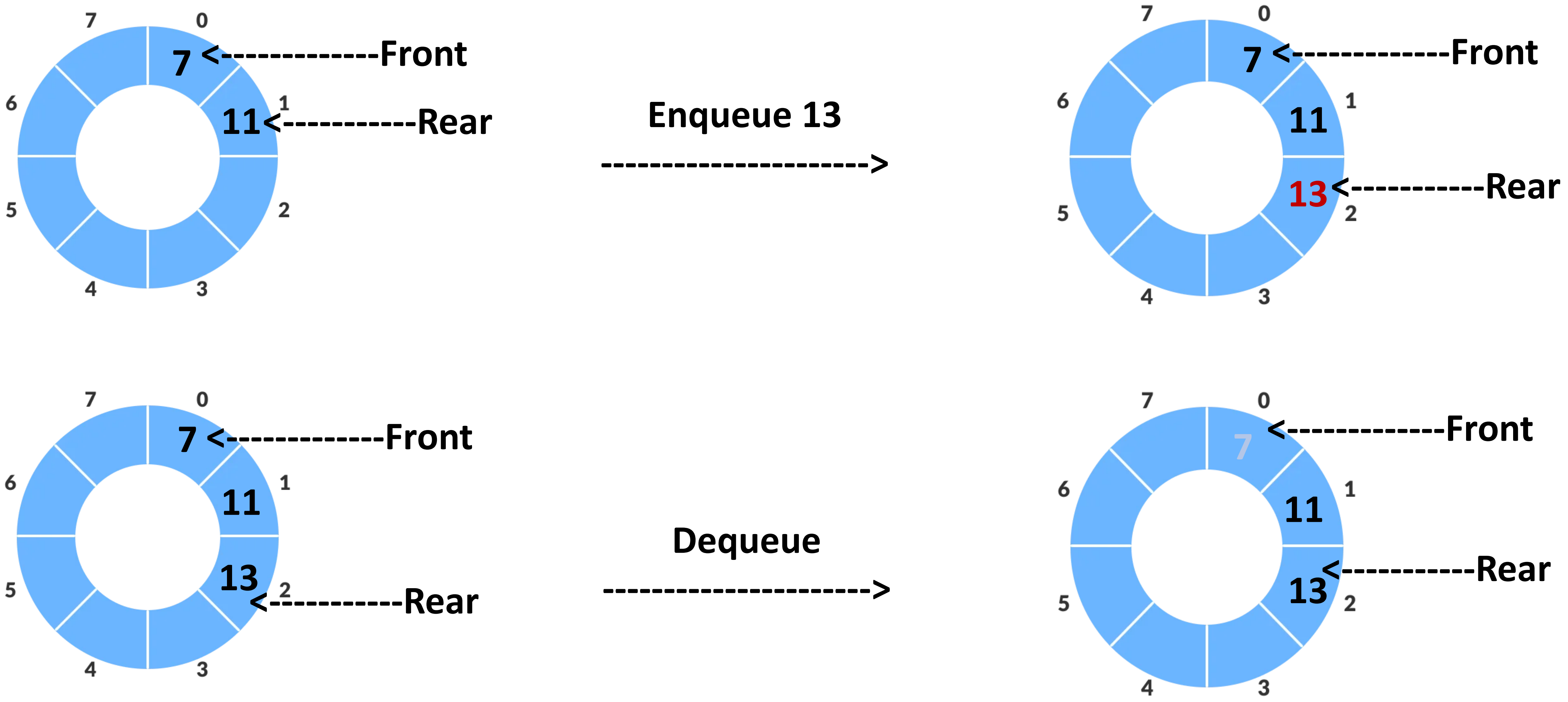
Enqueue 11

----------------------->



F- Front R - Rear

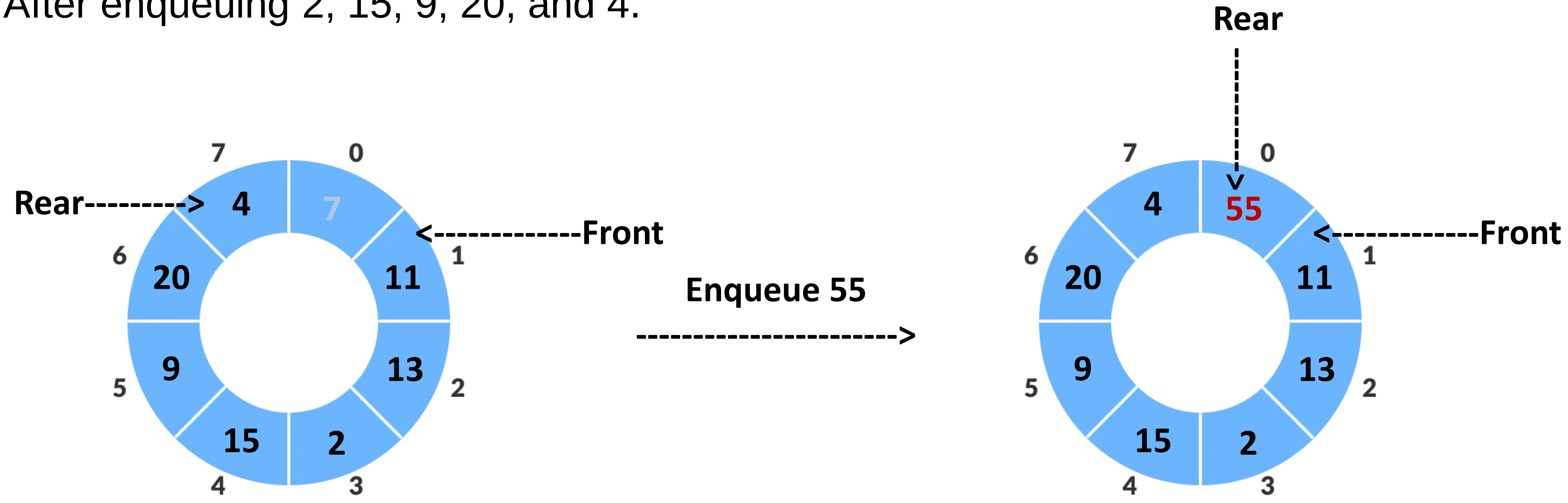
OPERATIONS OF A CIRCULAR QUEUE



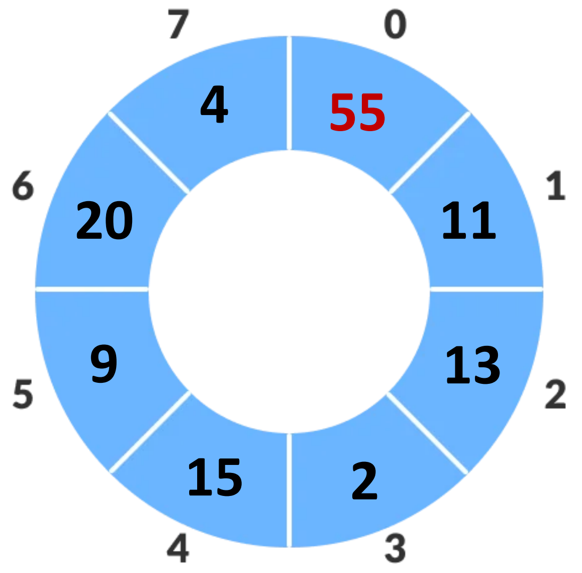
F- Front R - Rear

OPERATIONS OF A CIRCULAR QUEUE

After enqueuing 2, 15, 9, 20, and 4.



OPERATIONS OF A CIRCULAR QUEUE

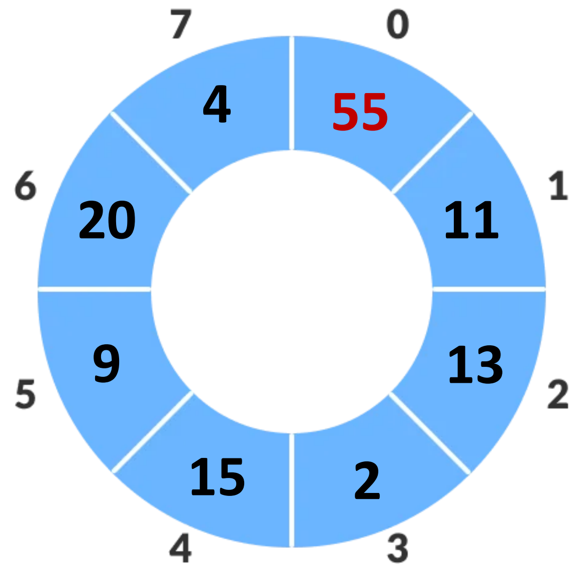


F = 1

R = 0

Enqueue 6

----------------------->



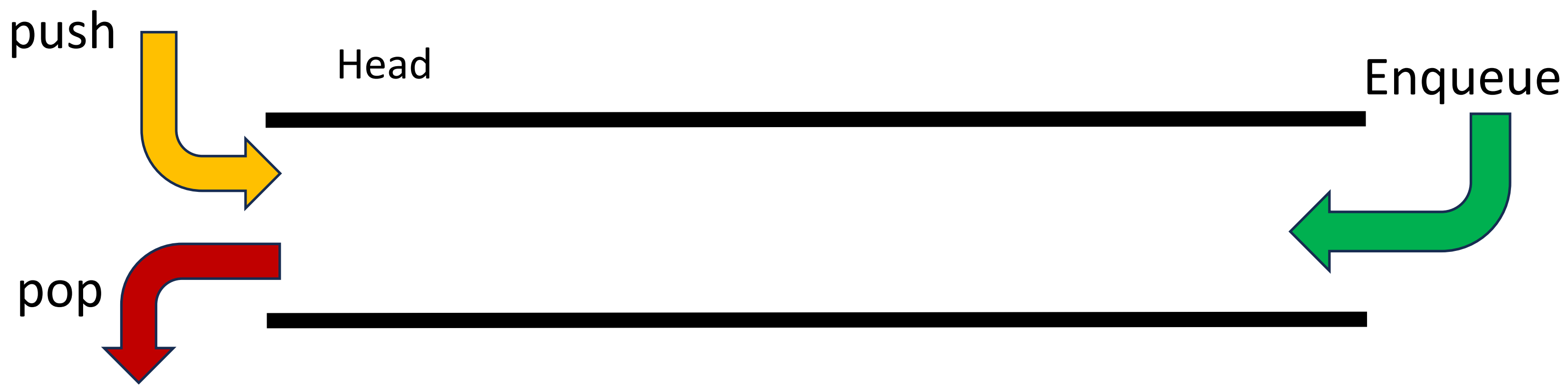
F = 1

R = 0

- Enqueuing 6 cannot be done. Queue overflow.

STEQUE

- A "steque" is a combination of a stack and a queue data structure.
- A stack-ended queue supports push, pop, and enqueue.
- In steque one end used to enqueue elements while the other end works as a stack, where we can push and pop both.



OPERATIONS OF A STEQUE

- Push:
  - Adds an element to the top of the steque.
  - After a push operation, the newly added element becomes the "front" of the steque.

- Pop:
  - Removes and returns the element from the top of the steque.
  - After a pop operation, the element previously below the removed element becomes the new "front" of the steque.

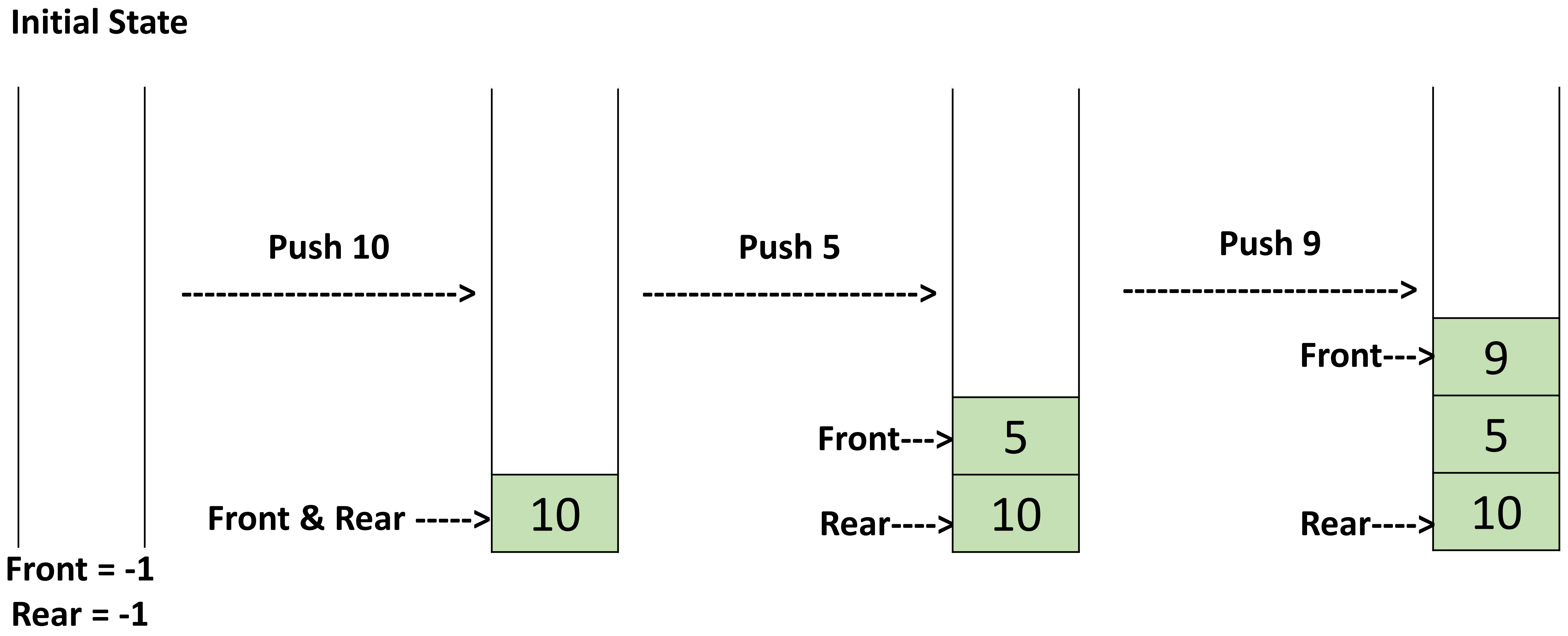
- Enqueue:
  - Add an element to the end of the steque.
  - After an enqueue operation, the newly added element becomes the "rear" of the steque.

OPERATIONS OF A STEQUE

- Peek
    - Return the value in the queue without removing it.
- IsEmpty()
- IsFull()

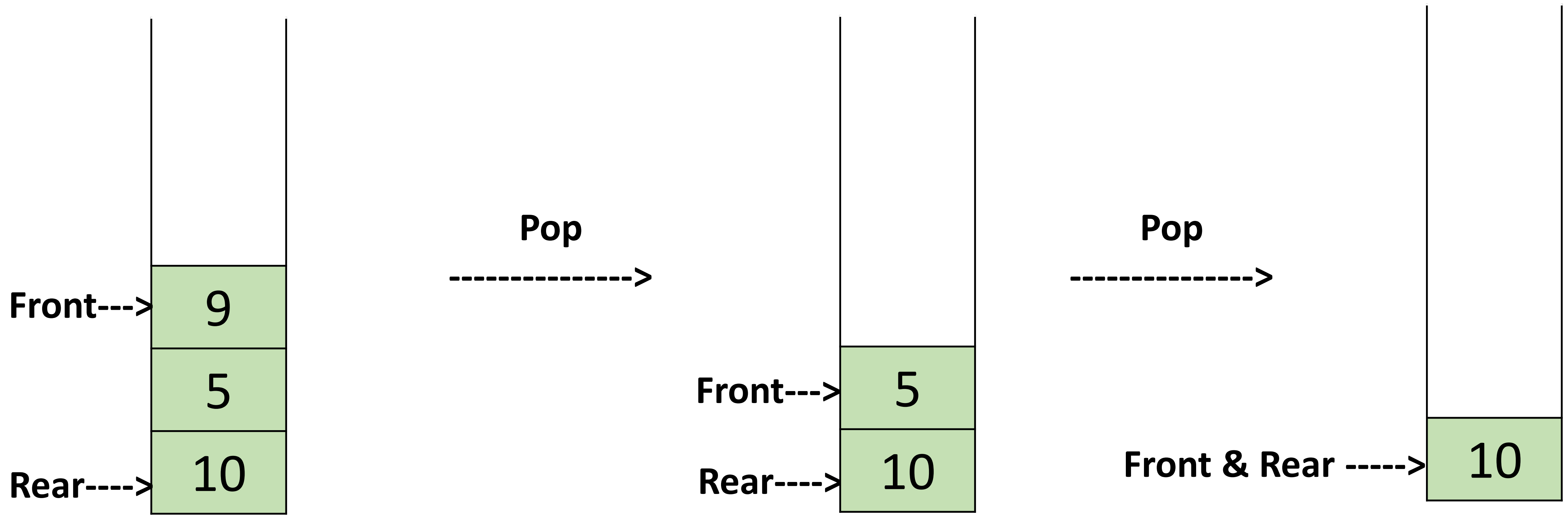
STEQUE

Push Operation



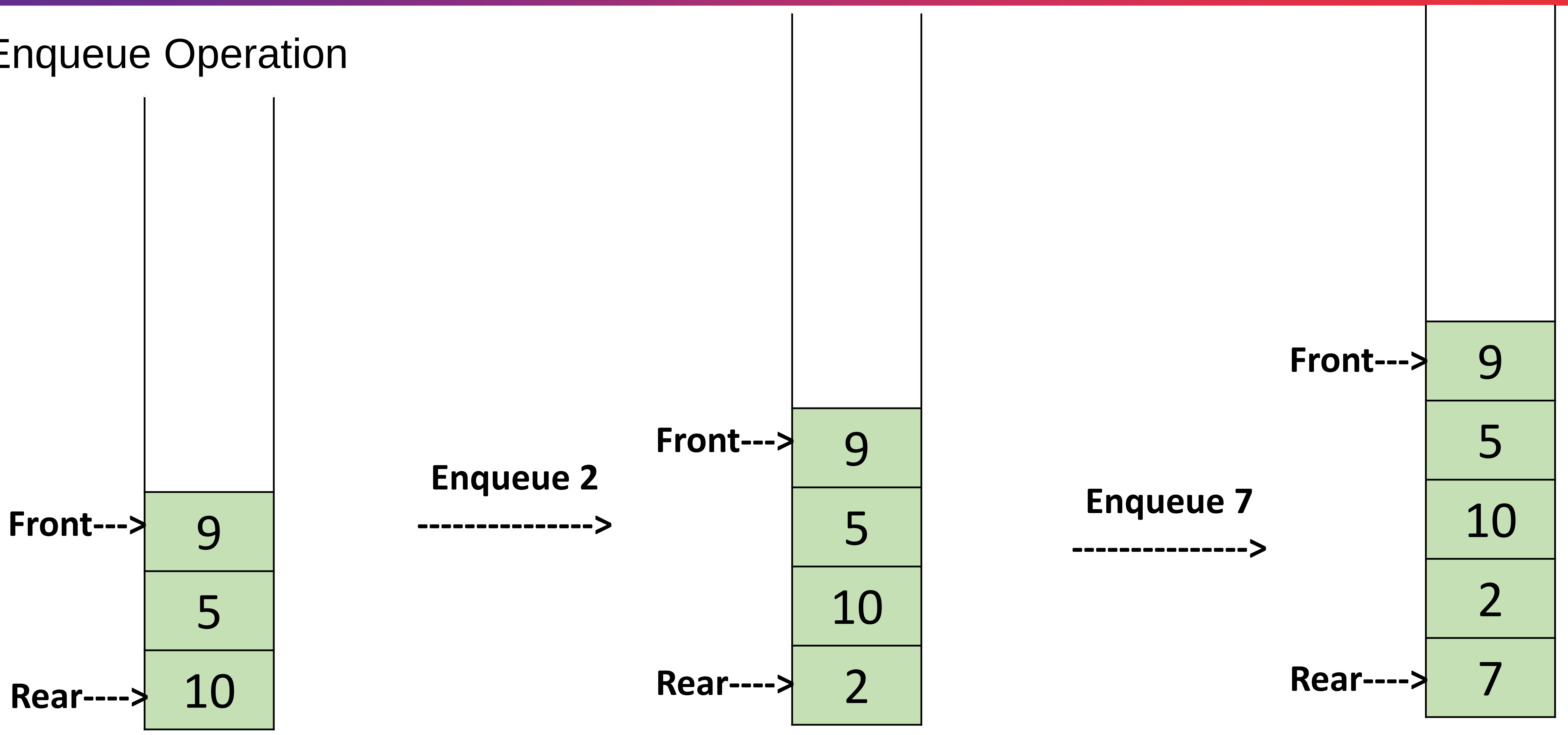
STEQUE

Pop Operation



STEQUE

Enqueue Operation



Activity

Illustrate the state of the steque after each operation.

- Push(5)
- Enqueue(11)
- Enqueue(8)
- Pop()
- Push(15)
- Pop()
- Pop()

SUMMARY

By now you should be able to,

- ILO1: Describe what is a FIFO queue and its basic operations.
- ILO2: Explain the applications of FIFO queue.
- ILO3: Implement FIFO queues using different data structures.
- ILO4: Discuss about the variations of the FIFO queues.

NEXT
Linked Lists