
[...Knowledge]: Stack & Queue!



Content

1. Stack

1.1. Implementation

1.2. Real-life implementations

2. Queue

2.1. Implementation

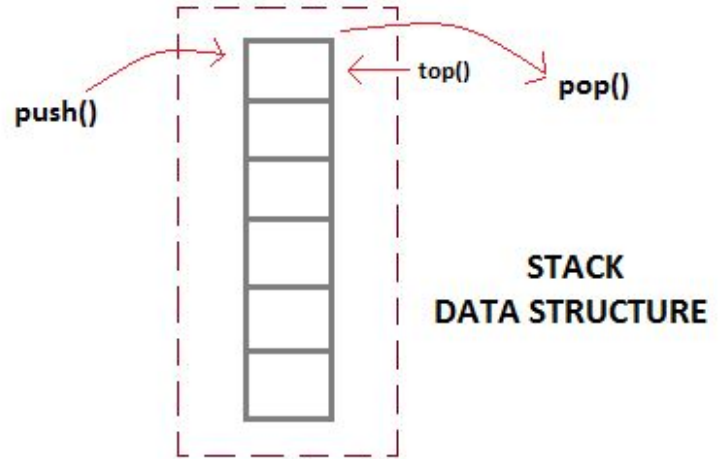
2.2. Real-life implementations

Stacks

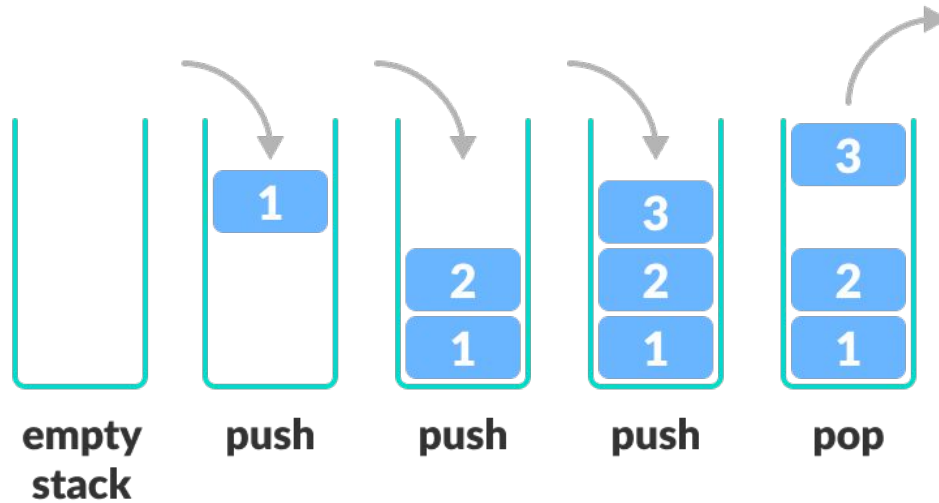


Stacks?

- Stack is a linear data structure of similar data type.
- Stack is a LIFO (Last in First out) structure.
- Stack has a bounded (predefined) capacity.
- Very fast $O(1)$

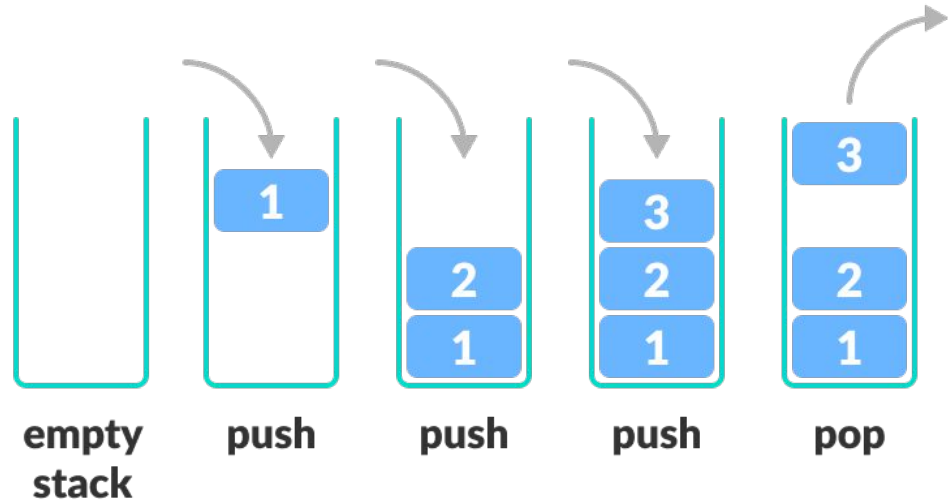


Stack: LIFO Principle

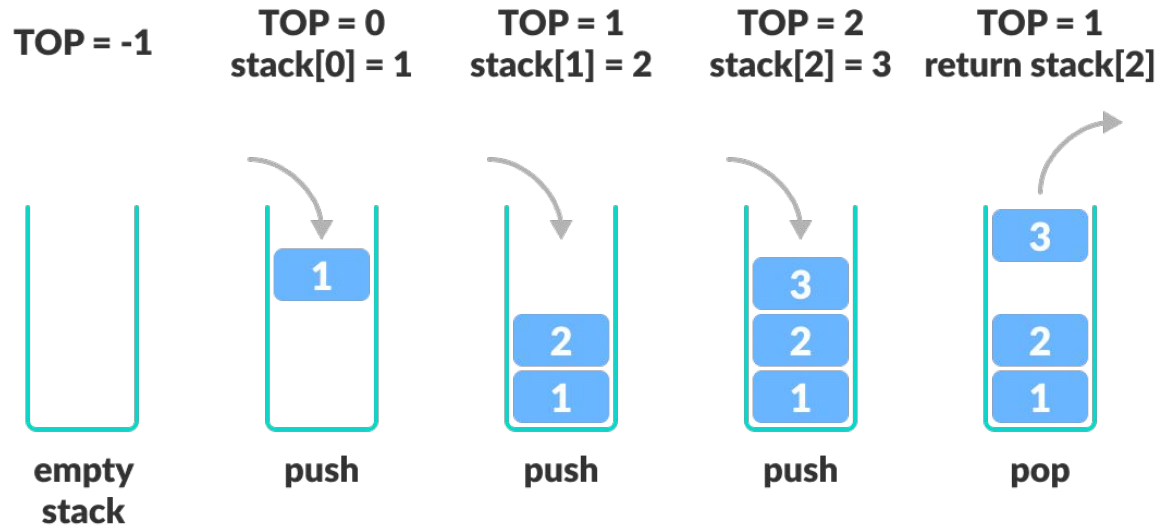


Stack

- Push
- Pop
- Peek



Stack



Stack:

Push

- Check if the stack is full or not.
 - If the stack is full, then print error of overflow. ❌
 - If the stack is not full, then increment the top and add the element. ✔️

Pop

- Check if the stack is empty or not.
 - If the stack is empty, then print error of underflow. ❌
 - If the stack is not empty, then print the element at the top and decrement the top. ✔️

—

AND NOW...
IT'S
Show Time

A vintage movie theater scene. A large screen at the front of the theater displays the text "AND NOW..." in a bold, sans-serif font, followed by "IT'S" in a smaller font, and "Show Time" in a large, elegant, cursive script. The screen is flanked by red curtains. In the foreground, rows of empty, dark-colored seats are visible, facing the screen. The theater's interior is dimly lit, with a greenish tint to the walls and ceiling.

Demo

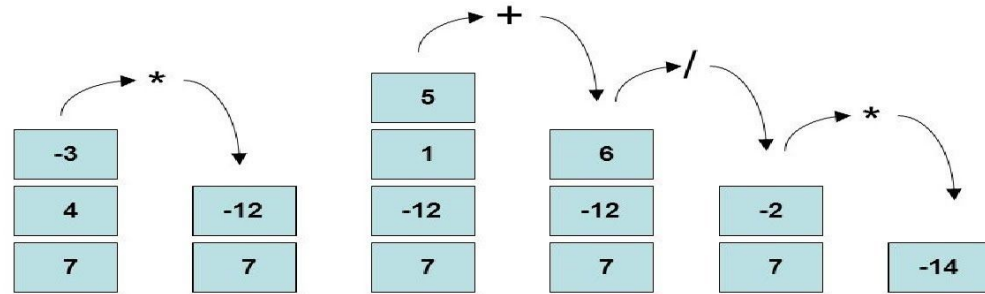
Stack: Real life implementations

- To reverse a word
- In compilers: Postfix notation
- In browsers: The back button in a browser
- JS: Call stack

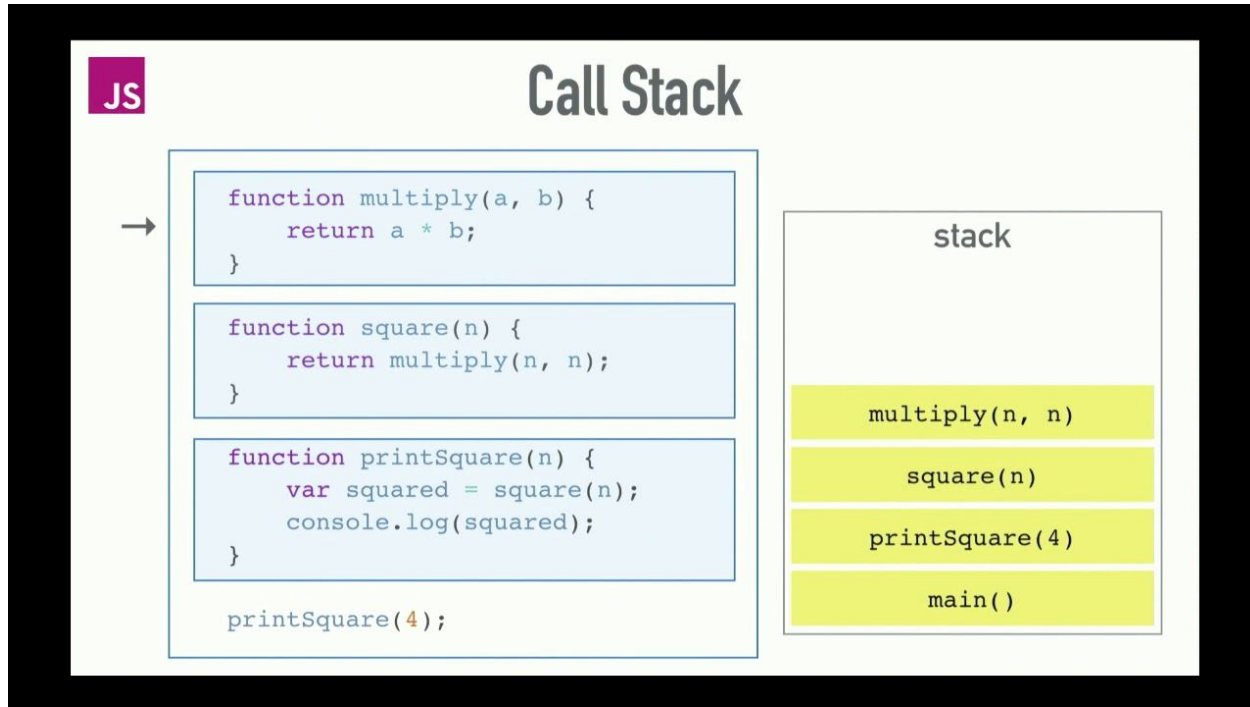
Stack: Real life implementations

Evaluating Postfix Expressions

- Expression = 7 4 -3 * 1 5 + / *



Stack: Real life implementations

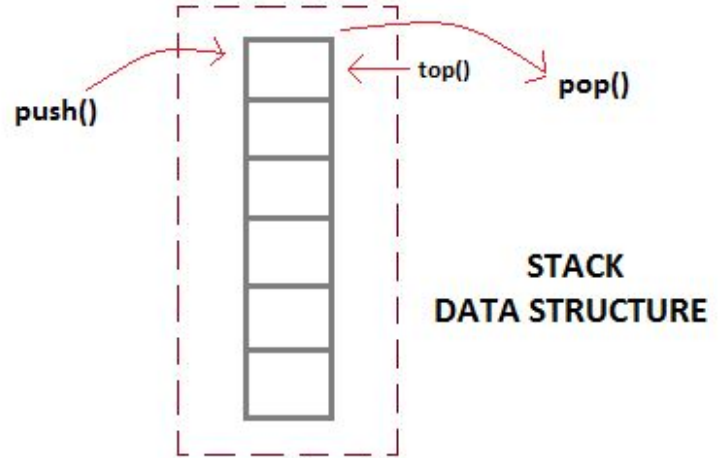


Queue



Queues?

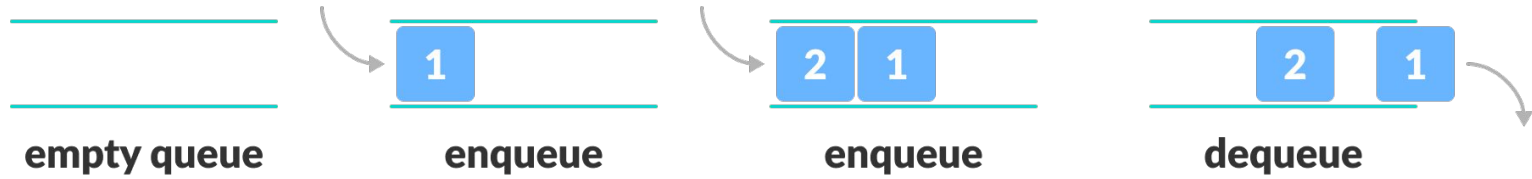
- Queue is a linear data structure of similar data type.
- Queue is a FIFO(First In First out) structure.
- Very fast $O(1)$
- Types: Simple, Circular and Priority



Queue: FIFO Principle

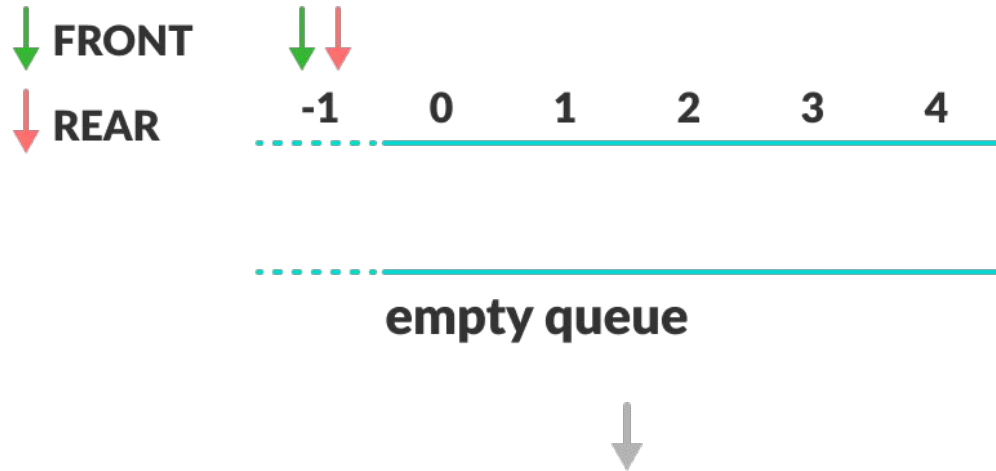


Queue



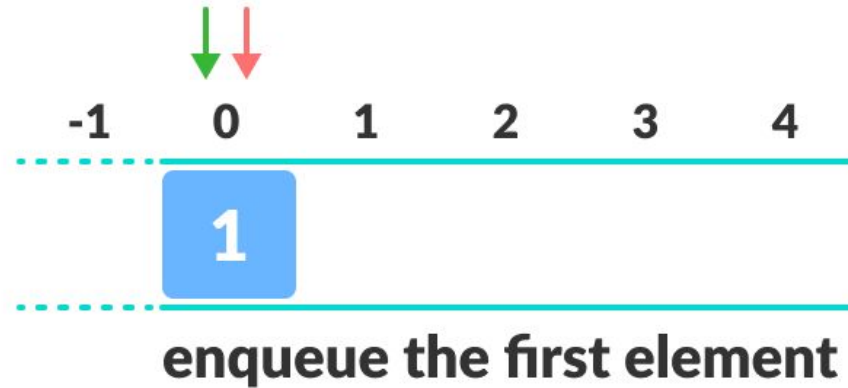
- Enqueue
- Dequeue
- Peek

Queue: Implementation

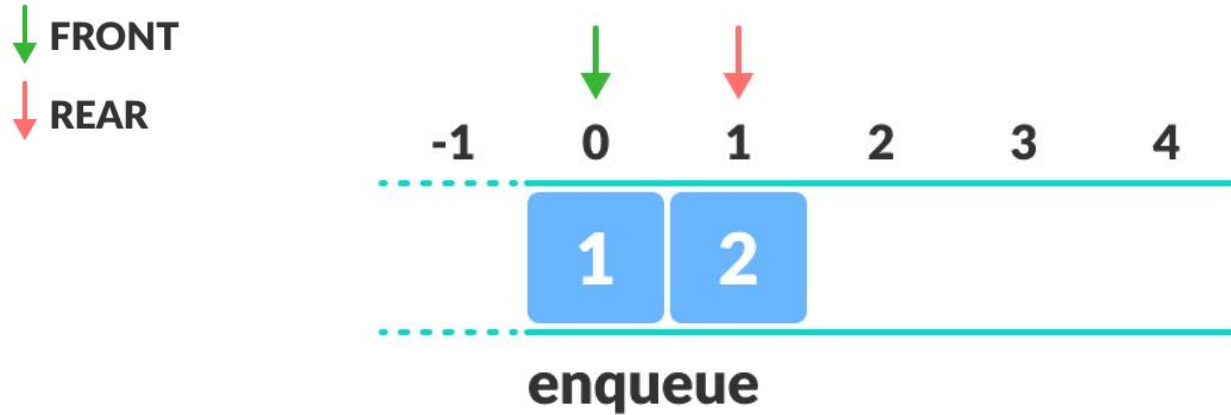


Queue: Implementation

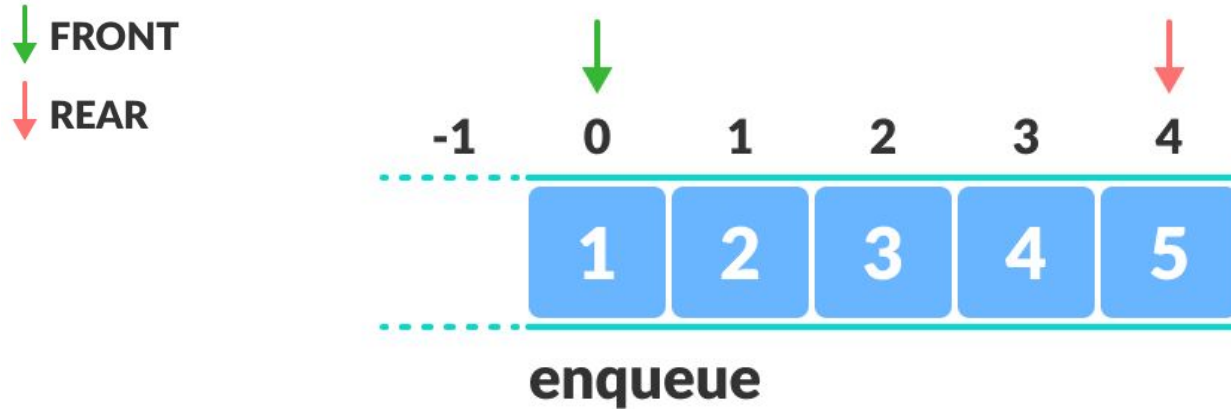
↓ FRONT
↓ REAR



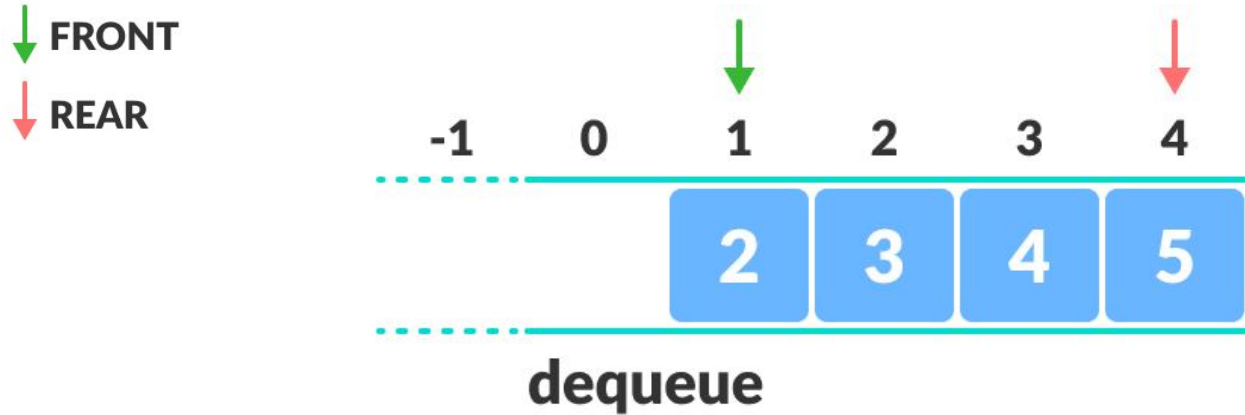
Queue: Implementation



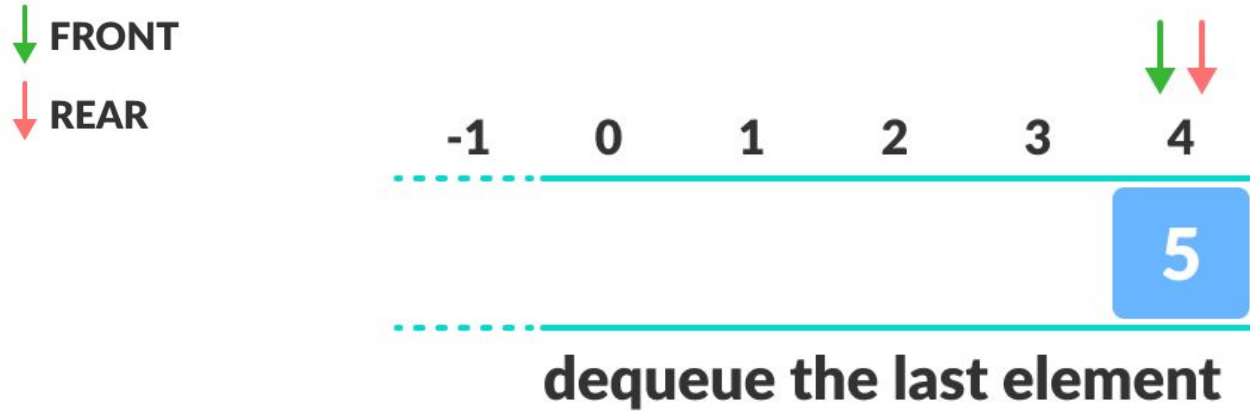
Queue: Implementation



Queue: Implementation



Queue: Implementation

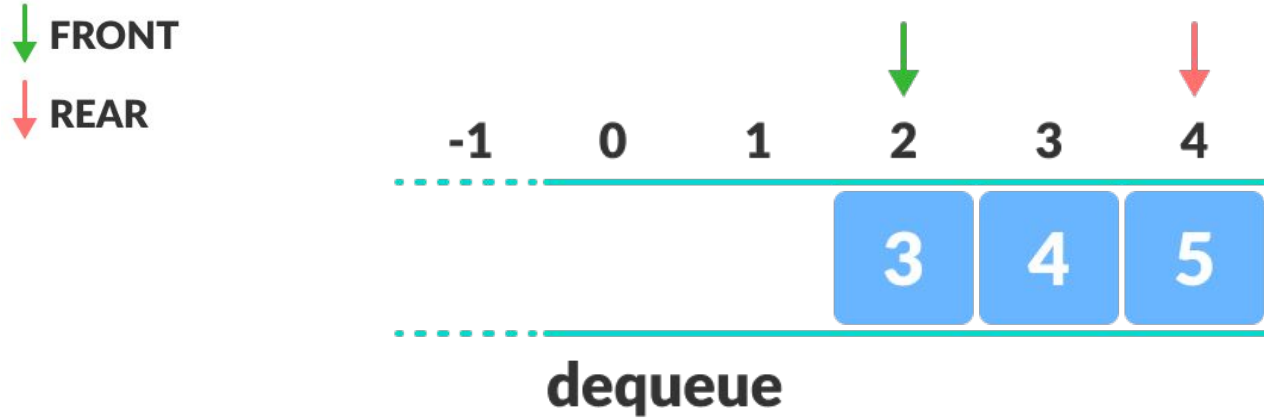


Queue: Implementation

↓ FRONT
↓ REAR

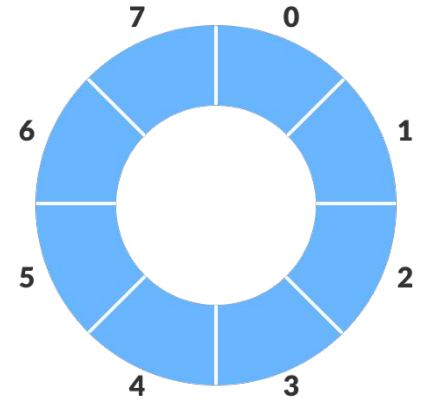
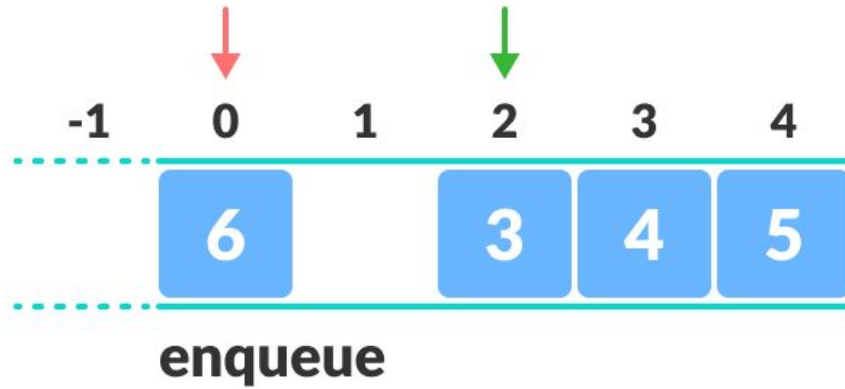


Queue: Limitation



Circular Queue

↓ FRONT
↓ REAR



Queue:

Enqueue

- Check if the queue is full or not.
 - If the queue is full, then print error of overflow. ❌
 - For the first element set value HEAD to zero. ✔️
 - Increment the tail and add the element. ✔️

Dequeue

- Check if the queue is empty or not.
 - If the queue is empty, then print error of underflow. ❌
 - for the last element, reset the values of HEAD and TAIL to -1. ✔️
 - Print the element at the head and increment the head. ✔️

—

AND NOW...
IT'S
Show Time

A vintage movie theater scene. A large screen at the front of the theater displays the text "AND NOW..." in a bold, sans-serif font, followed by "IT'S" in a smaller font, and "Show Time" in a large, elegant, cursive script. The screen is flanked by red curtains. In the foreground, rows of empty, dark-colored theater seats are visible, facing the screen. The overall atmosphere is that of a classic cinema.

Demo

Queue: Real life implementations

- CPU scheduling, Disk Scheduling
- Call Center phone systems
- Traffic system

—

Thanks!