



S
T
A
C
K

→ **Push $O(1)$**

→ **Pop $O(1)$**

→ **Peek $O(1)$**



Activate Windows
Go to Settings to activate Windows

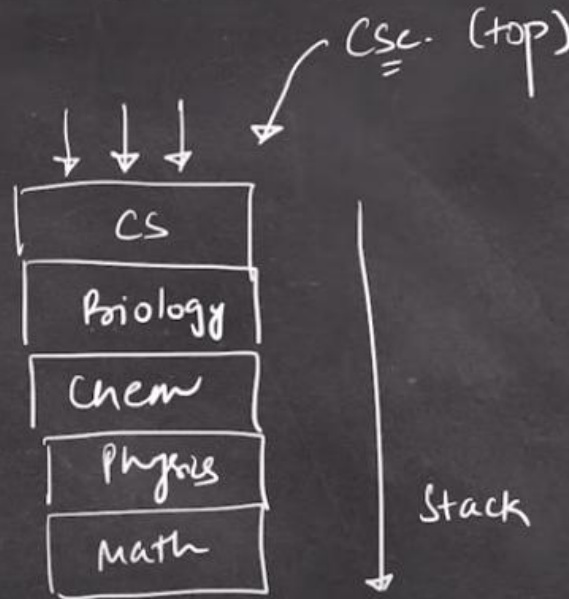
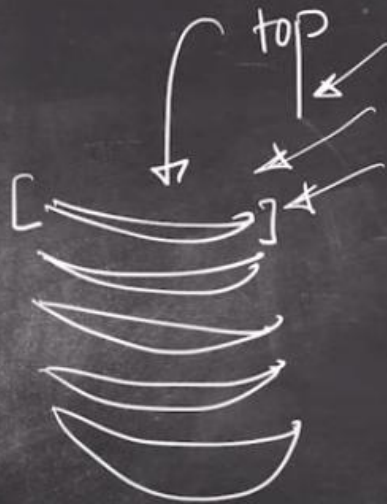


3:49 / 36:44 • Stack Introduction >

Scroll for details
↓



Real Life Examples



Activate Windows
Go to Settings to activate Windows.



STACK



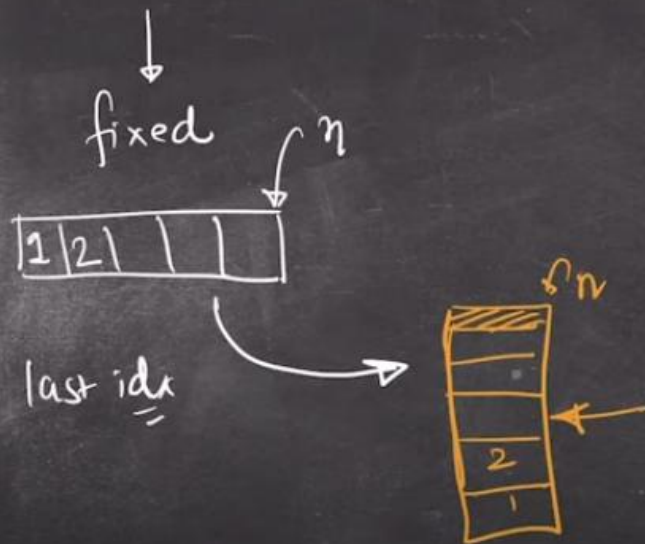
6:05 / 36:44 • Stack Introduction >

Scroll for details



Implementation

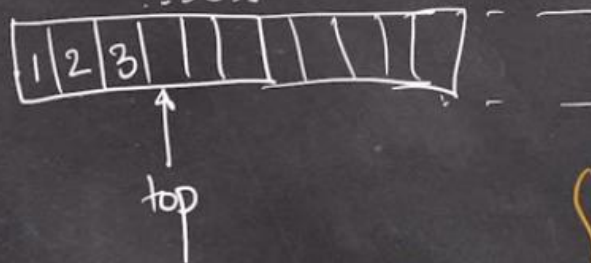
Array ^{1.} hectic



1. Stack full (we el == n)

ArrayList ^{2.}

variable



Linked List ^{3.}

variable



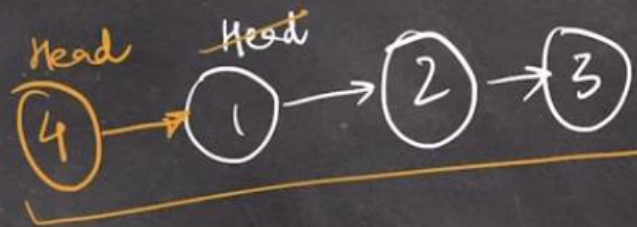
STACK

10:12 / 36:44 • Stack Introduction >

Scroll for details

Activate Windows
Go to Settings to activate Windows

APNA
COLLEGE



Activate Windows
Go to Settings to activate Windows



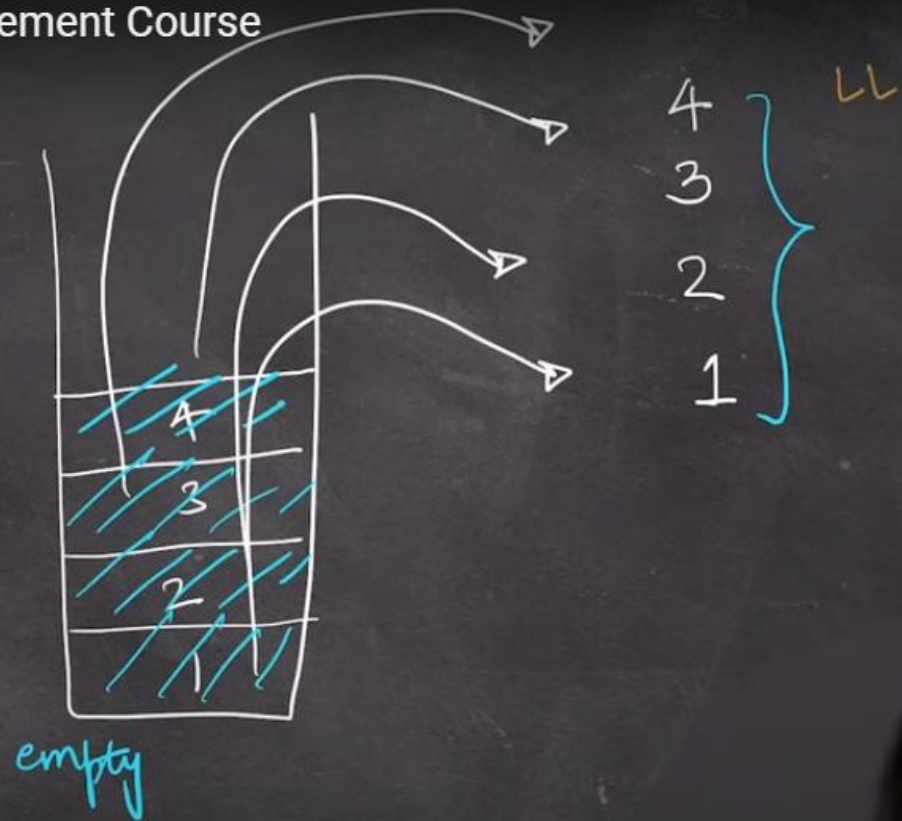
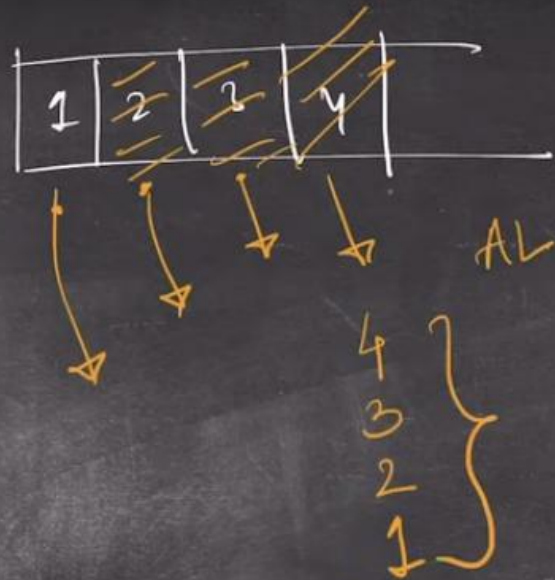
STACK



14:18 / 36:44 • Stack using Linked List > Scroll for details



Stack Data Structure in One Video | Java Placement Course



STACK

23:23 / 36:44 • Stack using ArrayList > Scroll for details

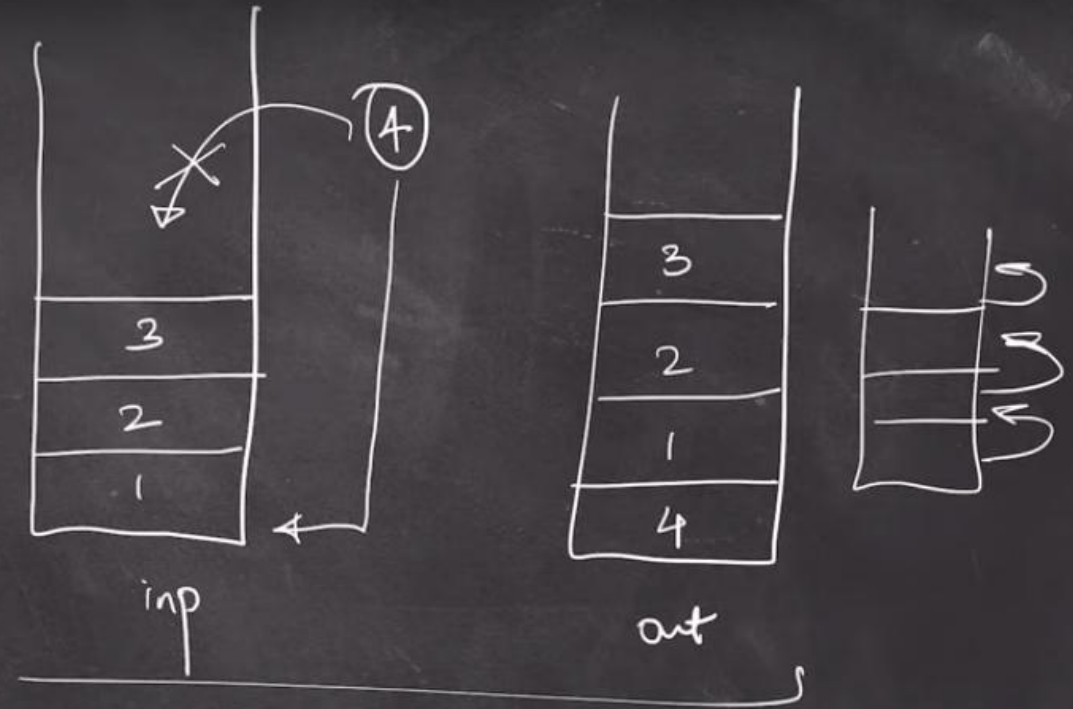
Activate Windows
Go to Settings to activate Windows

APNA
COLLEGE

Push at the Bottom of Stack

implicit

explicit



Activate Windows
Go to Settings to activate Windows

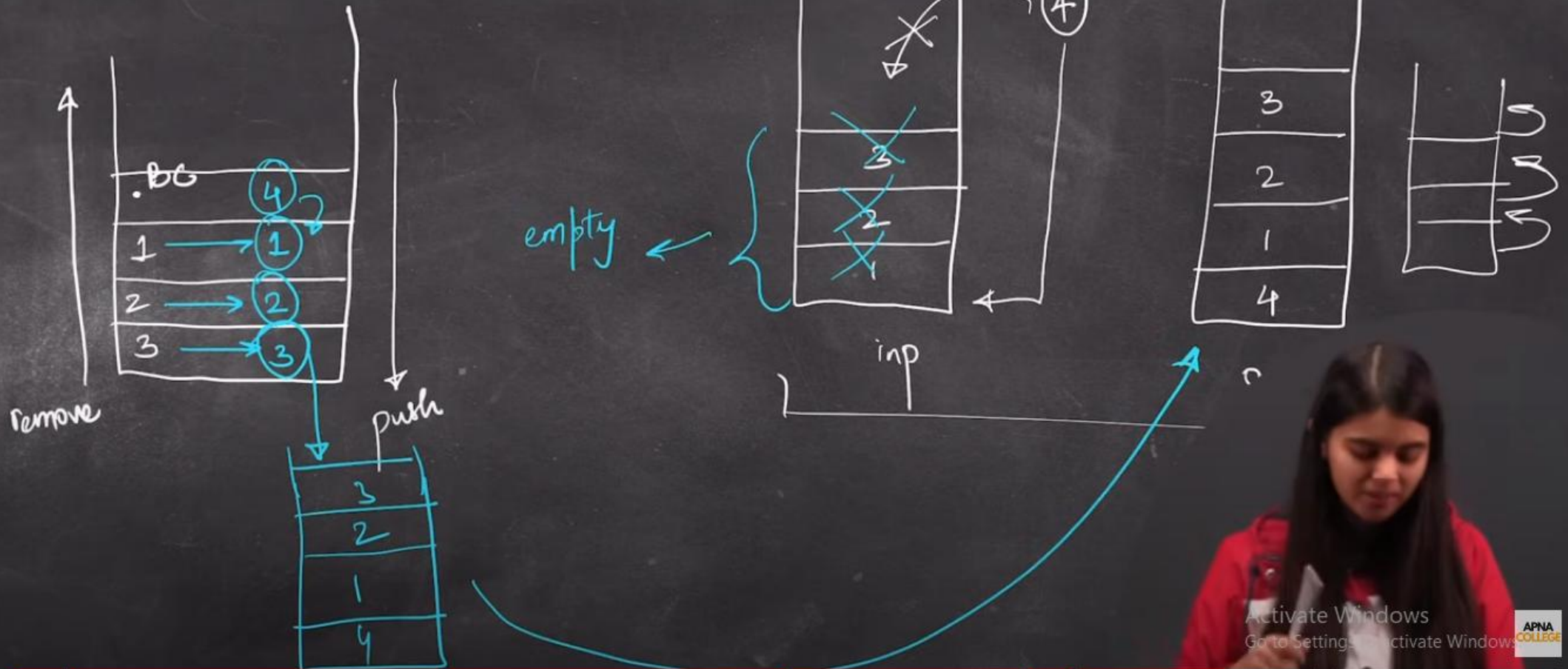


26:30 / 36:44 • Question1 (Push at Bottom of Stack)

Scroll for details



Push at the Bottom of Stack



STACK

28:43 / 36:44 • Question1 (Push at Bottom of Stack)

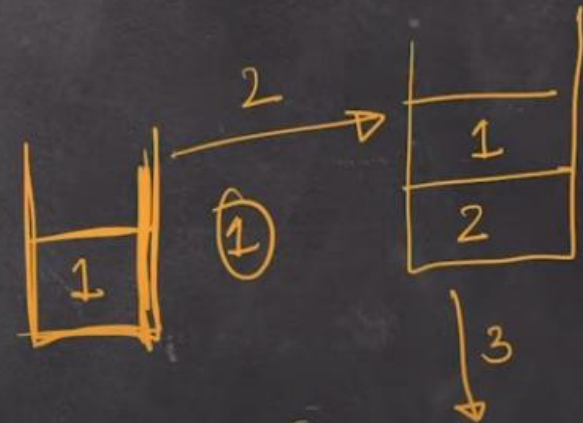
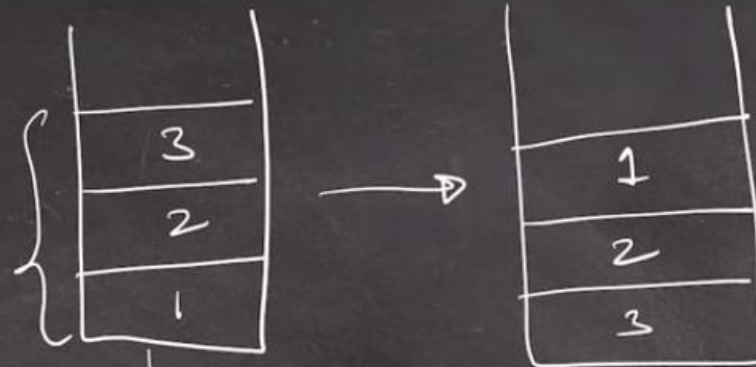
Scroll for details



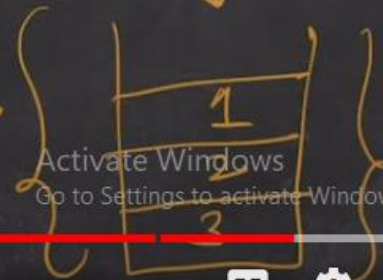
Activate Windows
Go to Settings to activate Windows.



Reverse a Stack

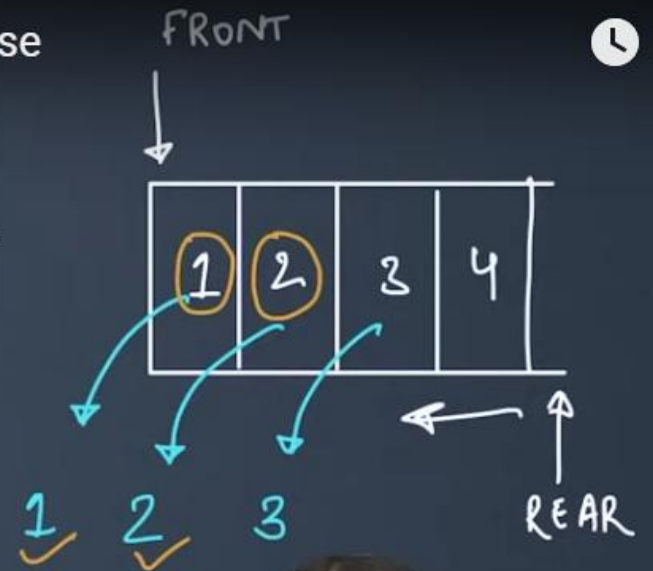
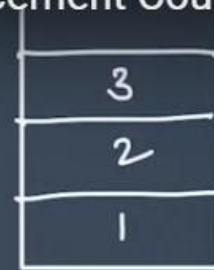


Reverse =



QUEUE

FIFO
First In First Out

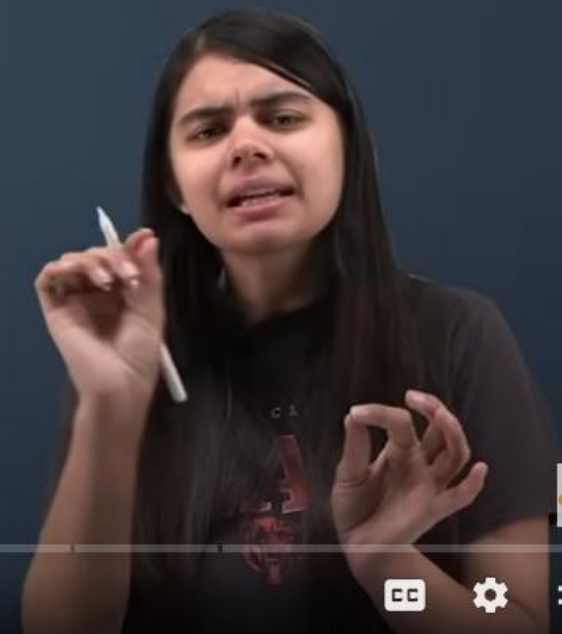
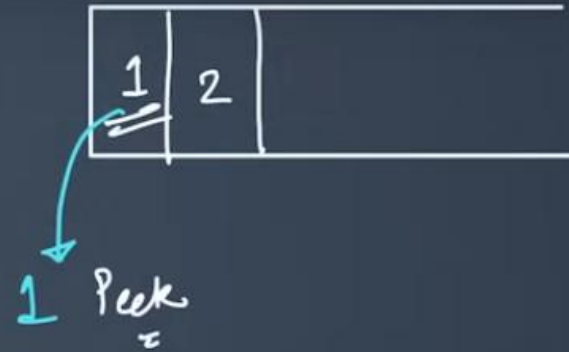


Operations

Enqueue Add → adding el

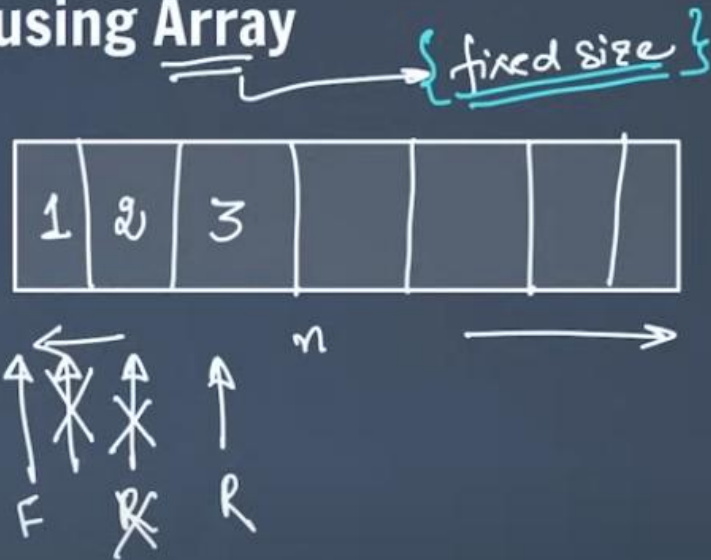
Dequeue Remove
Dequeue }

Front Peek



Implementation 1

Queue using Array



(i) $size = n$

⇒ ADD
full?

$F \Rightarrow 0$
 $R \Rightarrow \text{idx last el}$

$R = -1$
 $F = -1$ } →



Implementation 1

Queue using Array



(i) size = n

⇒ ADD
full?

F ⇒ 0
R ⇒ idx last el

$R = -1$
 $F = -1$ } →

add - $O(1)$
remove - $O(n)$
+ peak

Activate Windows
Go to Settings to activate Windows



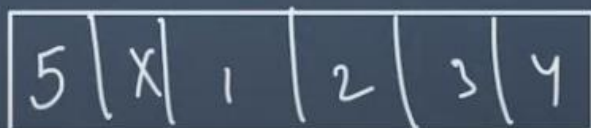
Implementation 2

Circular Queue using Array

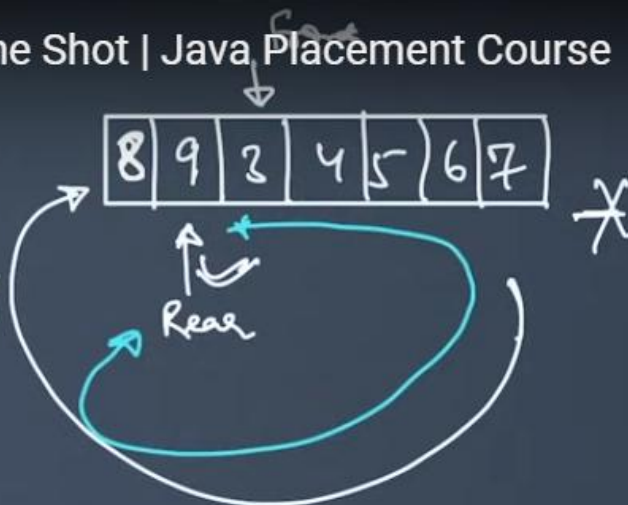
add $O(1)$
peek +
remove $\} O(n)$

add $O(1)$
remove
peek

size = 6



0
↑
Rear



8
9

full

$Rear + 1 == front$

Rear = -1
front = -1

Rear++

$$Rear = (Rear + 1) \% size$$

5

$$(5 + 1) \% 6 \Rightarrow 6 \% 6$$
$$\Rightarrow 0$$

Activate Windows
Go to Settings to activate Windows



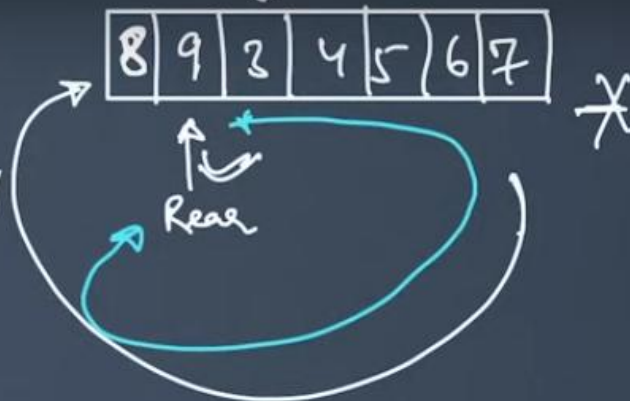
Implementation 2

Circular Queue using Array

add $O(1)$
peek +
remove $\} O(n)$

add $O(1)$
remove \uparrow
peek

size = 6



8
9

$Rear + 1 == front$

Rear = -1
Front = -1

Rear++

$Rear = (Rear + 1) \% size$

Rear++ X

$$\begin{aligned} 5 \\ (5+1) \% 6 &\Rightarrow 6 \% 6 \\ &\Rightarrow 0 \end{aligned}$$

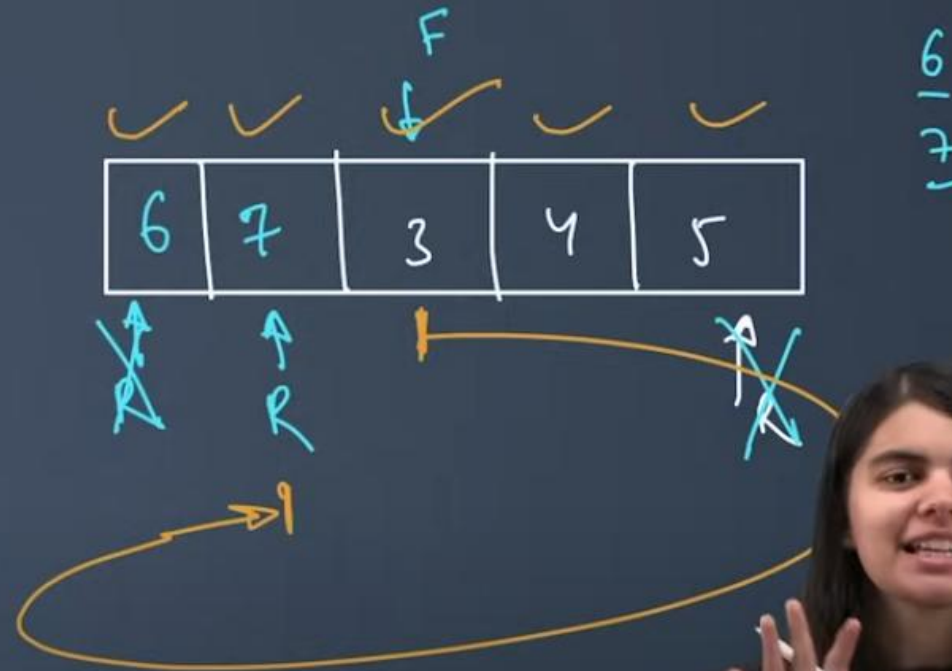
$(Rear + 1) \% size == front$

Activate Windows
Go to Settings to activate Windows

APNA
COLLEGE

Implementation 2

Circular Queue using Array



Activate Windows
Go to Settings to activate Windows.

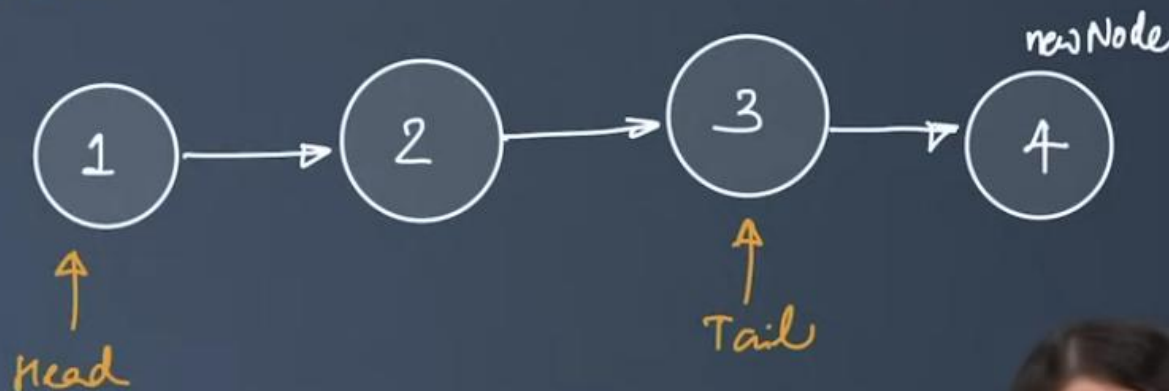


add



Implementation 3

Queue using Linked List



Activate Windows
Go to Settings to activate Windows

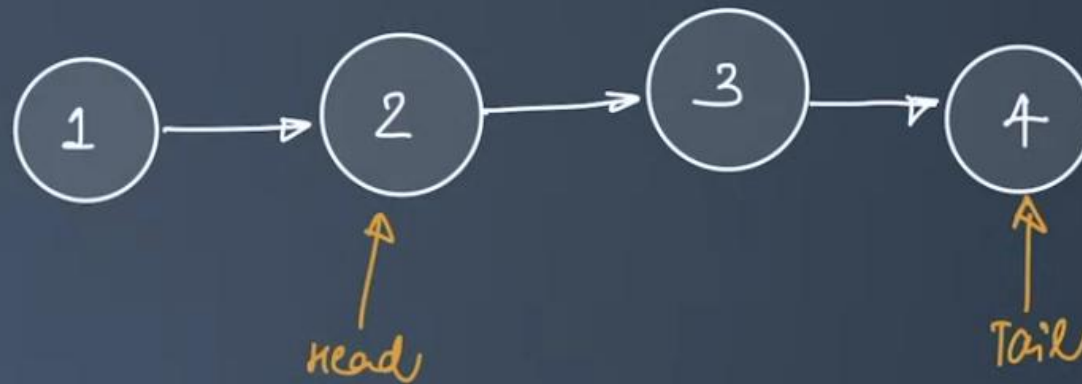


Implementation 3

Queue using Linked List

add

remove



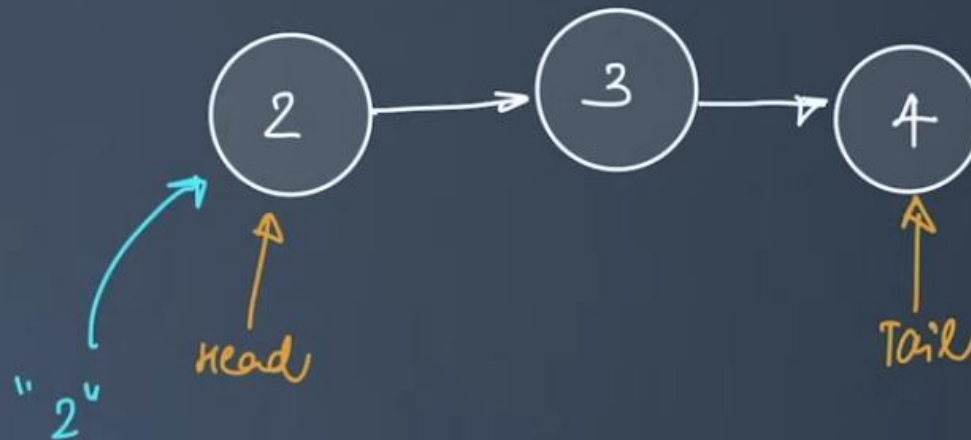
Activate Windows
Go to Settings to activate Windows



Implementation 3

Queue using Linked List

add
remove

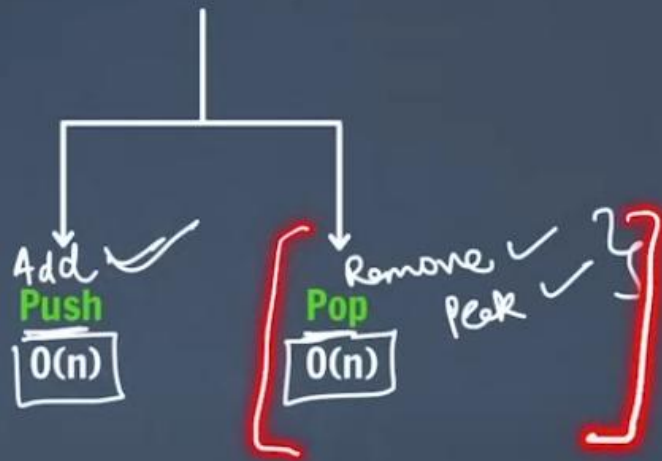


Activate Windows
Go to Settings to activate Windows.



Question

Queue using 2 Stacks



[illegible]

Complete Queue Data Structure | in One Shot | Java Placement Course

 **Apna College** ✓
5.41M subscribers

7.8K   Share ...

Java & DSA Course for Placement Windows
Apna College - 31 / 39 Go to Settings to activate
⏮ ⏭ ⋮

[illegible]


 Aurion Pro | www.aurionpro.com | info@aurionpro.com | www.aurionpro.com

...

Apna College - 31 / 39 Go to Set

Go to Settings to activate Windows.