June - 04

04 June 2020 20:59

Implement everything discussed in class.
Do Infix to Postfix with Brackets
Linked List implementation of queues. All the operations must be in O(1).

1. Infix to Postfix //Done
2. Queues
3. Array Implementation
4. Circular Queues

1 Infix to Postfix

1st → 3+(4*5)

3+(45*)

345*+

2nd → (3*4)+5

→ (34*)+5

→ 34*5+

fixed → ? Order of Operands is fixed

String → Left → LIFO (Stack)

1st 3+4*5 → 345*+

2nd 3*4+5 → 34*5+

To pada hai ushi precedence high

3rd

2+(3*4)-(5÷6)

ans = "234*+56÷-"

Stack: * , +

If the precedence of operator on the top of stack is high or equal then pop & add to resultant string

(2+(34*)-(56÷)

(234*+)-(56÷)

234*+56÷-

Queues → First In First Out → FIFO

Queues → First in

Why! ✓

Where? → Printer → Queue → CPU Disk Scheduling

ADT? → Insert → Enqueue → add( ) O(1)
→ Remove → Dequeue → poll( ) → delete & return
→ Peek() → only return
→ isEmpty()

Implementation

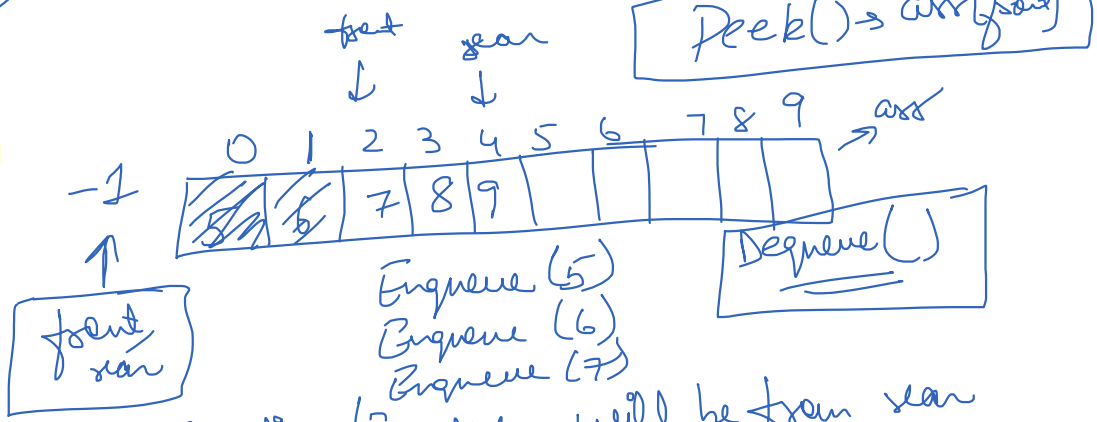
Array

LL

Stack

Peek() → arr[front]

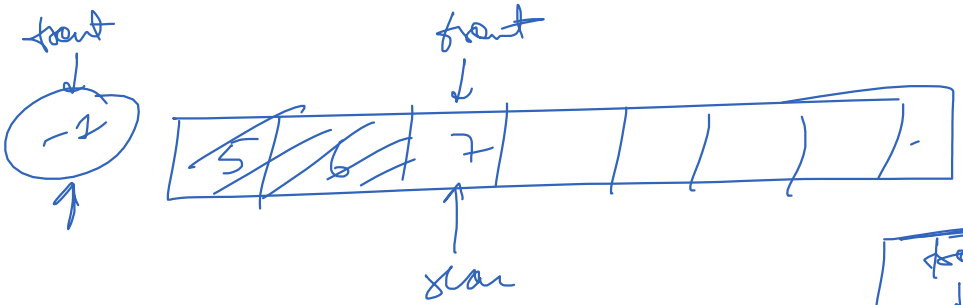
ArrayImp



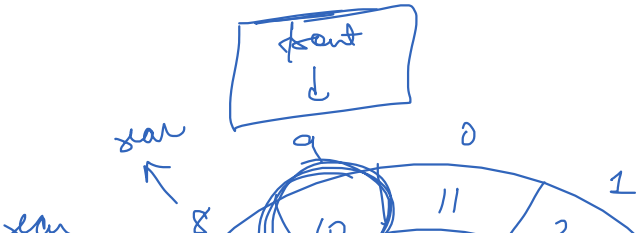
Enqueue (5)
Enqueue (6)
Enqueue (7)

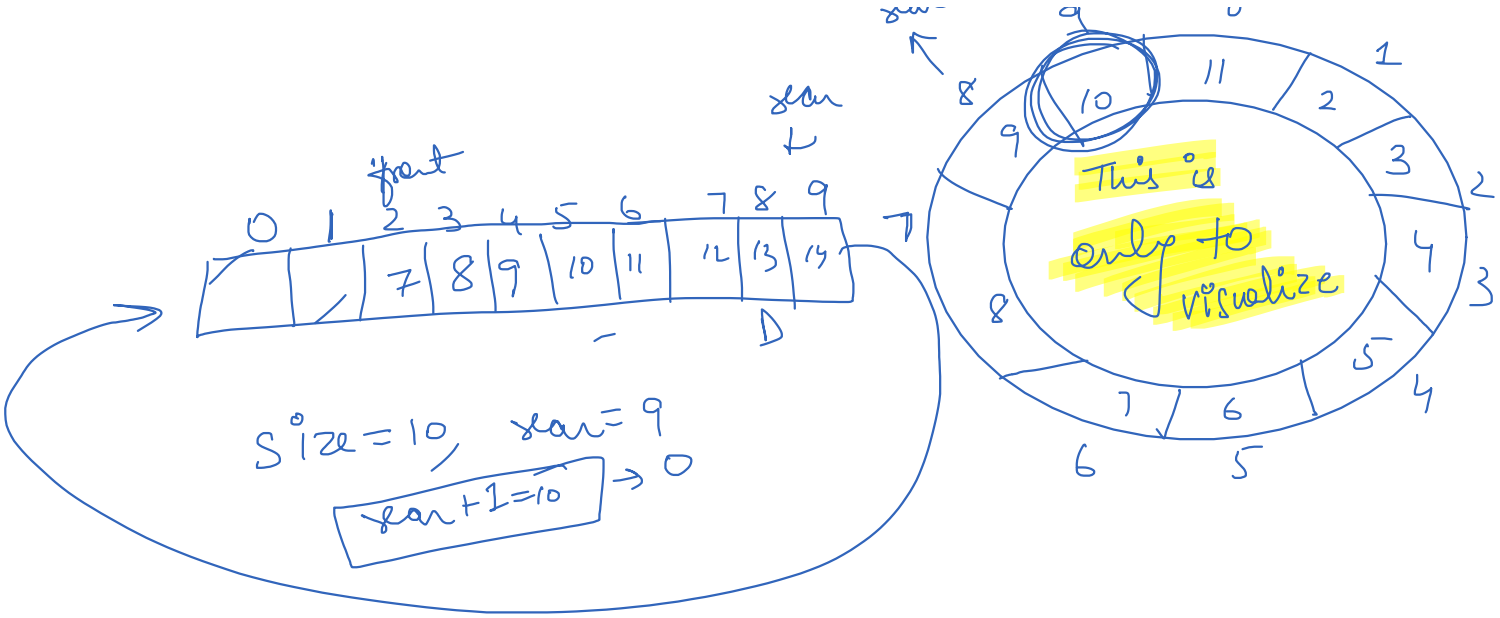
Insertion / Enqueue will be from rear

Deletion / Dequeue will be from front



(front+1) % size





(rear + 1) % Size → Using this I have connected last index to 0th index

If (rear + 1) % Size = front ⇒ full