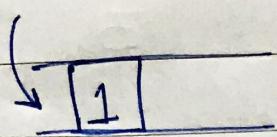


# Queue

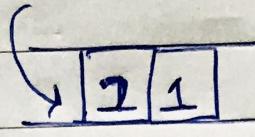
→ Linear D.S

→ First in First Out

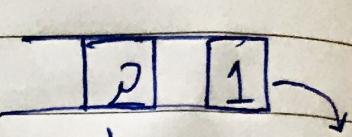
Empty queue



enqueue



enqueue



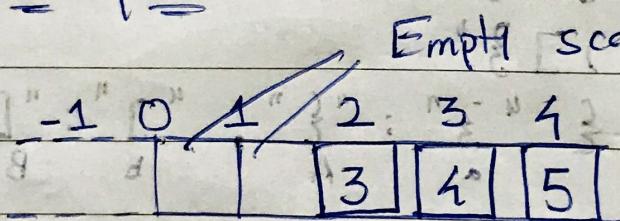
dequeue

Operation :-

→ Enqueue Operation :- Add element

→ Dequeue Operation :- Delete element

Limitation of Queue :-



Empty space

cannot be stored

Index '0' and '1' can only be used after the queue is reset when all elements have been dequeued.

Complexity :-  $O(1)$  for both enqueue & dequeue

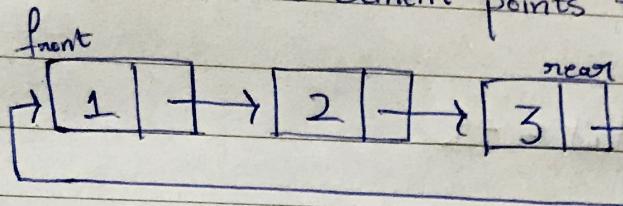
Application :-

- CPU & Disk scheduling
- Call centers

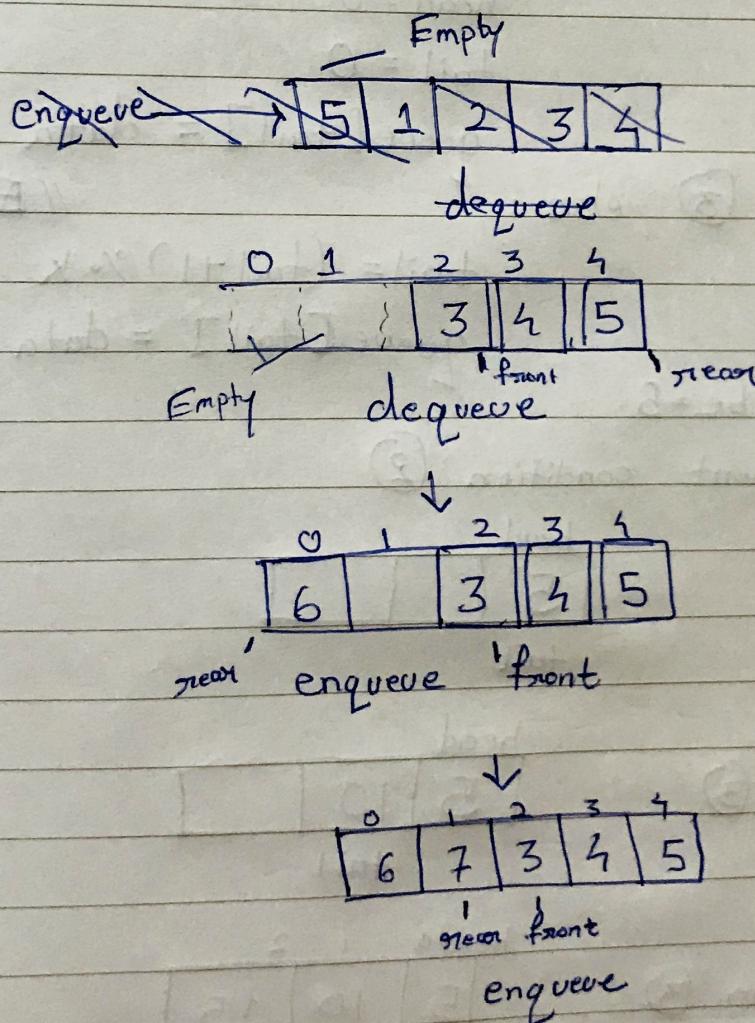
Types of queue :-

=> Circular Queue :-

Last element points to the first element



- Advantage :-
  - Better memory utilization
  - If last position is full and the first position is empty then, an element can be inserted



- Time Complexity :-  $O(1)$  for both enqueue & dequeue.
- Application :-
  - i) CPU scheduling
  - ii) Memory Management
  - iii) Traffic Management

## Enqueue :-

$k = \text{No. of elements}$

head = front of queue

tail = rear of queue

Let  $k = 3$

Base : head = tail = -1

def enqueue(self, data) :

① if  $(tail+1) \% k == \text{head}$ )

" Queue is Full "

② elif  $\text{head} == -1$ ) # First element

head = 0

tail = 0

queue[tail] = data

③ else :

// Element exist

tail =  $(tail+1) \% k$

queue[tail] = data

Let first element be = 5

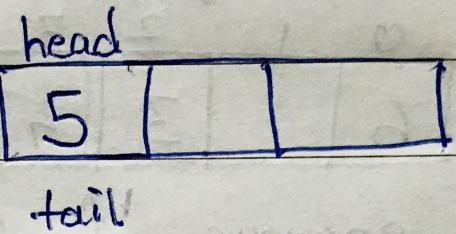
③ else :

$$\text{tail} = (\text{tail} + 1) \% \text{k}$$

$$\text{queue}[\text{tail}] = \text{data}$$

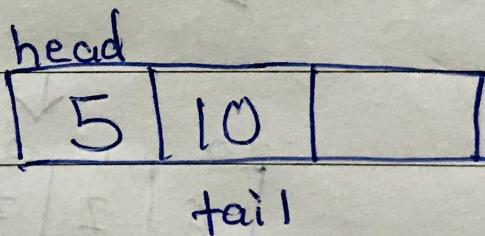
Let first element be = 5

For first element condition ②



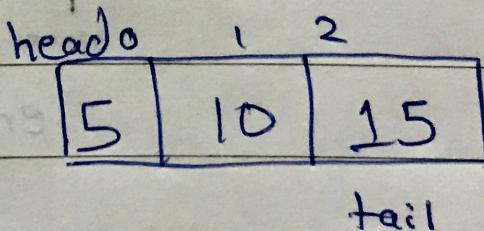
$$2^{\text{nd}} = 10$$

Condition ③



$$3^{\text{rd}} = 15$$

condition ③



$$4^{\text{th}} = 20$$

Condition ①

"Queue is Full"

$$\begin{aligned}\text{tail} &= (\text{tail} + 1) \% \text{k} \\ &= 3 \% 3 \\ &= 0\end{aligned}$$

## Dequeue :-

def dequeue(self):

① if  $\text{head} == -1$  # Queue is empty  
    "Empty"

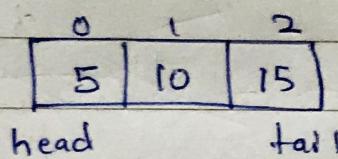
② elif  $\text{head} == \text{tail}$  # First element  
     $\text{head} = -1$   
     $\text{tail} = -1$

③ else:

$\text{head} = (\text{head} + 1) \% k$

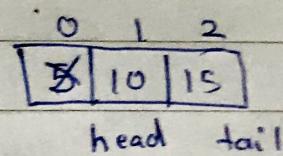
$$\begin{bmatrix} 0+1 \% 3 \\ \text{head} = 1 \end{bmatrix}$$

Our queue is :-



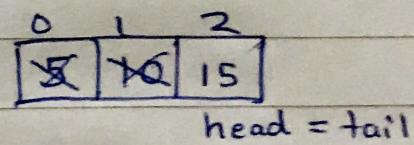
i) dequeue()

Condition ③ :



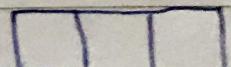
ii) dequeue()

Condition ③ :



iii) dequeue()

Condition ② :  $\text{head} = \text{tail} = -1$



iv) dequeue()

Condition ① : "Empty" [  $\text{head} = -1$  ]